WHY 'IN-THE-CLOUD' SCANNING IS NOT A SOLUTION

Maik Morgenstern, Andreas Marx
AV-Test GmbH, Klewitzstr. 7, 39112 Magdeburg, Germany

Email {mmorgen, amarx}@av-test.de

ABSTRACT

Currently, 'in-the-cloud' services are praised as the Holy Grail and the future of AV scanning. While such systems, built on both blacklisting and whitelisting approaches, can definitely increase detection rates and response times to new malware, this paper will show that current systems still have quite a lot of limitations:

- The implementations are not proactive, but reactive in nature, despite better response times to new threats.
- While detection rates are maximized (which looks good in test results), the risk of false positives is also increased.
- The results of 'in-the-cloud' scanning can be based on much more input data of both good and malicious files, but causes an additional performance impact on the client-, network- and server-side.
- Due to the time required to answer a query, only on-demand scanners and files which are executed are checked, but not all accessed files (as a 'traditional' on-access guard would work).

Our paper will also look at factors such as the limited caching of results, how data is transferred (e.g. via HTTP, HTTPS or DNS requests) as well as the privacy (e.g. what kind of data is submitted?), security (e.g. can responses be manipulated?), reliability and fault tolerance (e.g. what happens with a broken Internet connection?) issues of today's 'in-the-cloud' implementations by the different AV companies.

INTRODUCTION

With the ever changing threat landscape, security software vendors must develop new protection mechanisms and modify existing ones to be able to fulfil their task. With most of the innovations of recent years having been implemented on the client, e.g. behaviour-based (dynamic) detection and blocking, there is now a new approach that is based on the other side.

'In-the-cloud' services can usually be reached by the client over the Internet and are located on the vendor's infrastructure. This brings several advantages, such as much bigger databases on large server farms and instant updates in the cloud, without the need to deploy them to the user. But it may also bring problems. Additional points of failure are introduced when protection relies on a working Internet connection, and such services may not be able to solve the underlying problems of traditional detection mechanisms.

In this paper we will therefore look at what 'in-the-cloud' services are, what they can do and how they are implemented today. We will then look at the limitations and challenges that may originate from these points. Finally, some real-life experiences regarding such limitations and problems will be presented, showing that there are indeed challenges that need to be given serious consideration.

'IN-THE-CLOUD' SERVICES

This section looks at the basics of 'in-the-cloud' services. Why have they been introduced and why are they required to combat today's threat landscape? Are they required at all or is this yet another marketing hype or buzz word? Furthermore, there is the question of how they can help in fighting malware and which different variations are possible and have already been implemented. We will give an overview of today's 'in-the-cloud' services and possible future implementations.

Why 'in-the-cloud' services?

In May 2009 the number of malware samples in our collection reached nearly 22 million unique files. We see about one million new (unique) files every month and the numbers now seem to be growing even faster, after they stagnated a bit in the last year. A year ago there were only 10 million unique files and two years ago there were not even five million unique samples in our collection, see Figures 1 and 2. On the other

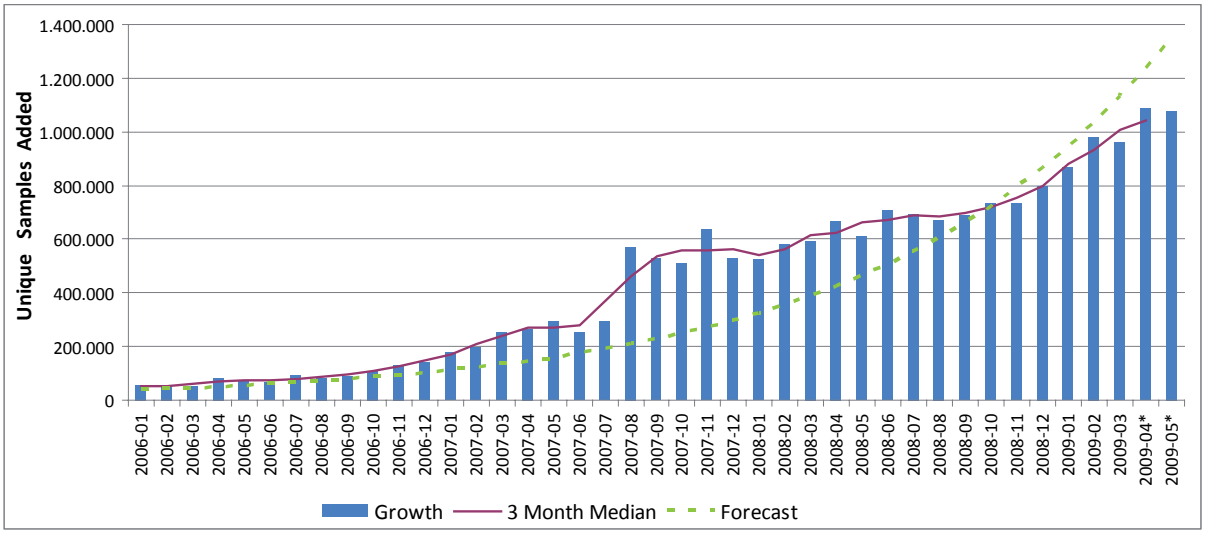

Figure 1: New unique samples added to AV-Test's malware collection.

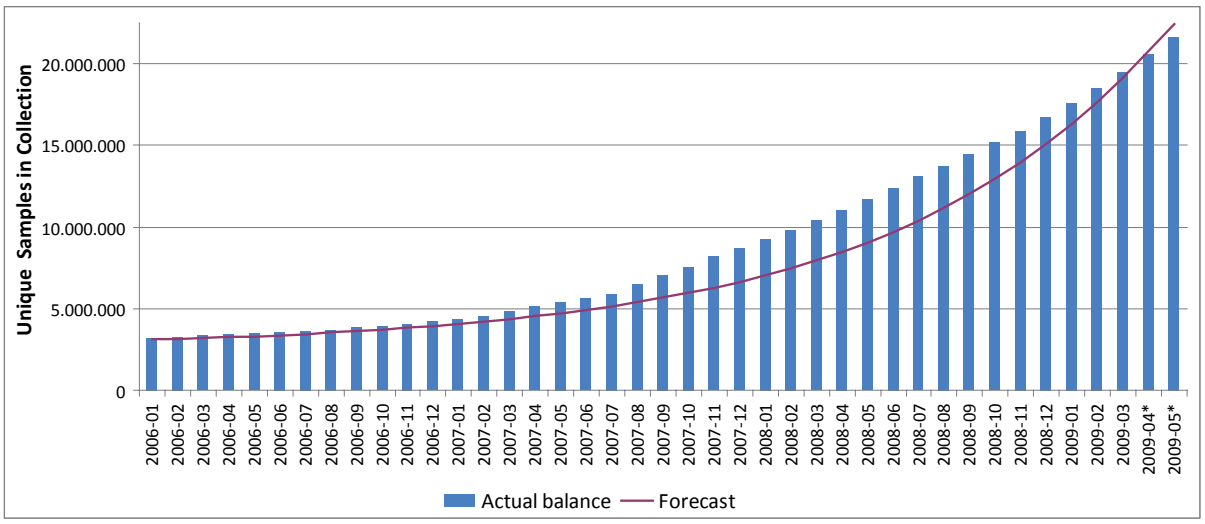

Figure 2: Total number of unique samples in AV-Test's malware collection.

hand, the lifetime of certain malware samples is often only days or even hours.

This all requires continual reactions and improvements from AV vendors to be able to protect customers. Several technologies have been developed during recent years, e.g. generic signatures, heuristics and behaviour-based (dynamic) protection mechanisms. Furthermore the number and size of signature updates has had to be increased by nearly all vendors.

For example, the regular DAT updates for *McAfee* products were 3.8MB in size on 1 January 2004. This includes the detection, cleaning and naming databases. One year later, the size had grown to 5.1MB, in 2006 it was 8.1MB and in 2007 there was an increase to 9.6MB. Right after this, the size increased to 19.1MB on 1 January 2008, and finally 66.6MB exactly one year later. Most other AV vendors had similar issues with the pattern file sizes. All started with less than 5MB in 2004 and are up to 91MB (*Panda* local signature file) or even 126MB (*Trend Micro* detection patterns of consumer products) as at 1 August 2009.

Interestingly, products like *Dr. Web* only require 15.4MB for the engine, definitions and on-demand scanner files with good malware detection rates. *ESET NOD32* is able to do the work with 18.2MB and *Avira* requires 22.9MB, both with very good detection score results (the sizes and the detection scores were all measured on 1 August 2009). It looks as if these vendors are reviewing their signatures databases again (after some time has passed) to check if more generic detection routines can be created.

However, with all the approaches described above, the workload and performance impact on the client gets greater and greater if the current technologies are used to fight the exploding numbers of malware. Yet, no user is willing to devote the majority of their computing power and memory to the 'simple' task of detecting and blocking malware. It is expected that AV software will consume little to no resources and leave all the computing power for the 'important' tasks. Furthermore, the signature updates have to be deployed to the user somehow and it is always necessary to deploy all existing signatures, even when only a very small fraction of them will actually be required on the specific system.

These are problems that can be addressed with 'in-the-cloud' technologies. The big signature databases can be stored on one central server and accessed through the Internet, instead of storing all the hundreds of megabytes of signatures on each individual client (on disk and partly in memory). Additionally, no updates have to be deployed to the user, since the data in the cloud is constantly being updated and always delivering up-to-date answers. The goal is to reduce the performance impact on the client and provide faster updates.

Furthermore, the in-the-cloud approach enables new or extended services and features. Bigger databases can be used, to both detect malware as well as to prevent false positives. Besides this, the most current disinfection for a specific piece of malware can be loaded from the cloud. Detailed information about files can be collected and tracked in order to heuristically determine suspicious files and identify trends. This leads to reputation systems. The relevance of certain malware could be determined, e.g. by measuring the prevalence, which can be used to prioritize samples.

'In-the-cloud' services do have the potential to take some performance impact from the client, provide faster updates and may add reasonable new and extended functionality to existing AV software. Details of possible approaches and implementations will be discussed in the next part of this paper.

Implementations of 'in-the-cloud' services

As mentioned above, there are different areas in which 'in-the-cloud' approaches can be useful. Depending on the purpose, the implementations can and will of course differ. A basic implementation is given in Figure 3, which shows how *McAfee* describes its *Artemis* technology to its customers. The set-ups of other vendors can be considered to be similar.

The basic idea is to have an object (in most cases a file) on the client that needs to be classified somehow (e.g. malware, clean or unknown), so that it can be handled or ignored by the security product. If it cannot be classified locally, a fingerprint, parts of the object or the whole object can be transferred to the cloud, which will then return a result that can be processed by the software on the client.

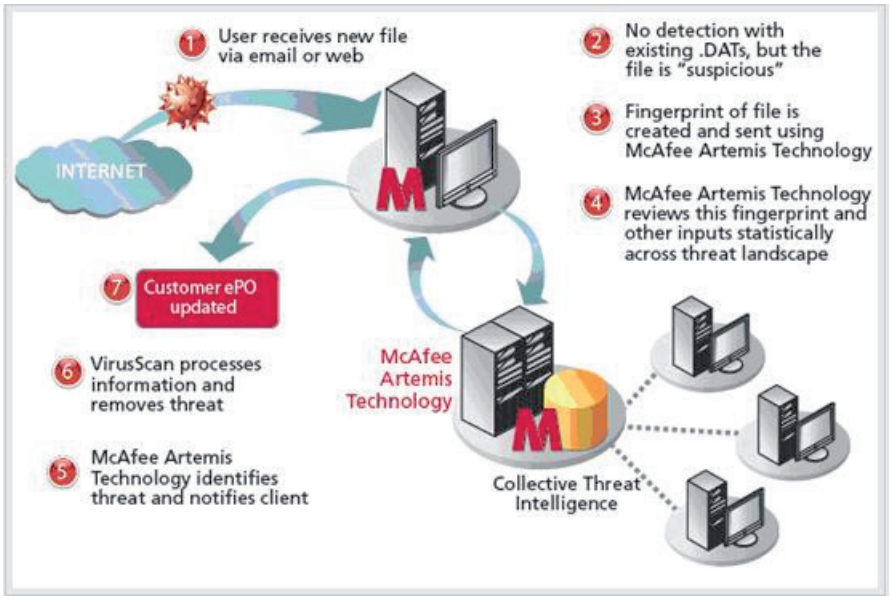

Figure 3: Technical brief: McAfee Artemis Technology [1].

The easiest implementation is to use fingerprints (e.g. MD5 hashes) of the objects on the client. This data is then submitted to the cloud which returns a decision as to whether the object is known or unknown. If it is known it can be classified as known bad (malware) or as a known good object. This is a classic reactive system and resembles the traditional signature-based detection, in this case with the fingerprint being the signature. This approach essentially moves the databases into the cloud, but doesn't add anything substantial to the old-fashioned static reactive detection systems. The actual sample still needs to be available to the vendor, so he can make an analysis and feed the decision to the cloud. The main advantages are bigger databases and instant updates in the cloud, instead of deploying every single update to the user.

In addition to those very simple hash matching approaches, it is also possible to create somewhat more proactive systems. Statistics, e.g. prevalence and first seen date, of the objects (or their fingerprints) that are submitted to the cloud can be used to draw conclusions. It is also possible to transfer more than just a fingerprint, e.g. information such as file size, used runtime packer, unusual PE structure, imports and exports and the like. This information can be used to create proactive detections with heuristic approaches. Furthermore, it can be extended to ultimately lead to a reputation-based system. The advantage of this approach is the more generic functionality which doesn't require access to the actual object. The information sent from the clients to the cloud may already be enough to come to a definite decision.

Open questions are, where the cloud is located, what data is transferred to it and how it is transferred. The answers to these questions depend on the functional range that is implemented.

The normal approach would be to put the cloud servers at the vendor's site, which is then reachable via the Internet, so it can support all the reactive as well as the proactive functions described above. Another choice would be to install a local cloud server at the customer's site, e.g. to protect a corporate network. A local server would not take advantage of the instant updating in the cloud (since the updates would still have to be deployed to the local cloud) nor can it be used to draw conclusions based on statistical input, simply because the input data is too small and limited. An advantage that could be introduced is the usage of custom rules or signatures to allow or deny certain objects on the clients. Dedicated hardware would be required in the case of a local server which is an additional cost factor. Judging from that, the usual implementation is probably the cloud at the vendor's site which is accessible via the Internet. However, we will also consider local clouds in our further thoughts.

The final question concerns what data is transferred and how it is transferred. It has already been mentioned that basically fingerprints, characteristics or the whole objects can be transferred. The first two options will probably be the most used but it is also imaginable that whole objects might be transferred without the vendor's knowledge. Of course, the user must have agreed to send it and no further privacy issues should be raised. The result that is received from the cloud basically comes down to the states 'known bad', 'known good' or 'unknown'. Future implementations might report a specific risk level instead. The administrator or user can then decide what risks he wants to take, depending on the environment or introduction vector of the sample. In addition to this, detailed information about the object can be received which may include a description of the object or removal routines and the like, to clean a system of the object when it is detected as malware.

Transfer of the data is done in various ways using TCP/IP connections or UDP packets. Simple HTTP requests are possible, both encrypted and not encrypted. Other implementations use DNS as an application layer to send their requests and receive results. It is also possible to mix the various options, depending on the purpose of the request. Simple hash matching requests are probably better sent via DNS, while more complex requests might benefit from using HTTP(S).

LIMITATIONS AND CHALLENGES

The above describes the basics of 'in-the-cloud' services and how they can be and are implemented today. We will now look at the problems that still exist even with 'in-the-cloud' services, or that can arise from them. The first part looks at theoretical aspects while the second part lists several real-life experiences.

Theoretical aspects

Some limitations and challenges originate from the way the 'in-the-cloud' services are implemented and used. Other limitations lie in the nature of the approach. Both types will be reviewed in this section.

Not all of the following limitations apply to all possible implementations of 'in-the-cloud' services, instead they give an overview of the things that could be a problem. Some of these concerns were mentioned by Peter Ször at the CARO Workshop 2009 in Budapest [2] and some others were expressed by Dave Cole [3].

- *Still static scanning 1*: The problems of static scanning have not been addressed, just moved from the client to the cloud. Signature databases are still growing, and they will eventually become too large for the cloud as well. According to Moore's law, the computing power doubles every 18 months, but the number of malware doubles every 10 to 12 months (and it gets faster). What then?
- *Still static scanning 2*: In order to perform a static scan, the vendor needs to know about the object. The technique is still reactive, just as before. Server-side polymorphism (where every target PC gets an individual unique version of the malware) remains a problem. Future implementations must not only look at full file hashes, but need to submit further information about the threat to the cloud. For example, when a file is unknown (based on a cryptographic hash), the cloud server might request further information, like the size, the type, the structure etc.
- *Updates still take time*: While it is not necessary to deploy updates to the user, the cloud needs to be updated. When considering the point directly above, the analysis of an object takes time and therefore it still takes time until the detection can be inserted in the cloud. Instant protection is not possible this way.
- *Updates have an instant effect*: As soon as information is in the cloud, it is being used on the client. When an erroneous signature is fed to the cloud, it has an instant effect and could cause false positives or false negatives. Also, QA on the client side is no longer possible – there is no choice as to whether or not to use an update if it is in the cloud.
- *Performance issues don't go away*: Some performance problems are solved, however others can be introduced. The client still needs some computing power to decide whether an object needs to be checked in the cloud. Also answers from the cloud may take a while. What happens to the object until the answer arrives?
- *Network issues 1*: What if the network connection fails? Is all the protection gone? How robustly will the implementations react to slow connections (dial-up)? What about spoofed answers? What about malware that disrupts the connection?
- *Network issues 2*: How many requests can the cloud servers handle? How easy is it to perform a DDoS attack using the product or special attack scenarios?
- *Single point of failure*: What happens when the cloud is not available or gives inconsistent answers? Is there still some kind of protection? An easy target for attacks?
- *Abuse or disclosure of information*: The malware could query the cloud to know whether it is blacklisted yet. The information that is being sent (both request and answer) may contain private data.
- *Privacy*: Which information is allowed to be transmitted? There may be different laws in different countries.
- *Vulnerabilities in the cloud*: In [4] we showed that a lot of vulnerabilities exist in security software; the cloud service could be affected as well. While it doesn't have to actually process the objects, it still needs to process some input data, which may come from untrusted sources.
- *Caching of results*: It may be infeasible to query the same object every time it is accessed, so some kind of caching will probably be used. Remediation of false positives can become a problem then, because the false detection may be cached for a while.
- *Offline detection*: This problem is linked with the point above. If the client is able to operate offline as well, then it needs some sort of caching. This may be especially relevant in the case of system cleaning, when the user prefers to be offline rather than online (because the malware could then still control the PC).
- *Manipulation of reputation systems*: Large botnets may have the power to manipulate reputation systems or at least distort statistics.
- *QA of reputation system/proactive detections is problematic*: Automated systems are prone to errors and could be manipulated as pointed out in the comment above, leading to erroneous decisions of the cloud.

Current and new 'in-the-cloud' security providers should look at this list and check what can be done to avoid such issues.

Real-life experiences

As a test centre, we review quite a lot of different products on a daily basis. Some important aspects related to the testing of 'in-the-cloud' solutions have already been published by AMTSO [5]. These include, but are not limited to, the two major issues of test environments: they are no longer controlled and reproducible and they cannot be sealed off for security.

The question is whether the theoretical aspects listed above are actually relevant and appearing in reality. Well, they do occur and a few of them are listed below. But we will also look into the good points of our findings. We don't include any vendors' names, since the technology is still new and we concede that there will be some teething problems. However, when we review the same technology next year, we will be more critical and will publish further details. Of course, all vendors will have had advance warning, so they can start working on the outlined issues.

- *Database synchronization*: During some tests we saw that different 'in-the-cloud' servers seem to have stored different data (the data was not updated for some time) or even incomplete information (only a part of the database was accessible at a specific server, so no real transaction security was in place). Therefore, the information stored on the different servers was inconsistent, but the user was not able to notice this. This happened multiple times for two different AV vendors, both of which confirmed and fixed the reported problems.
- *Networking issues*: Even with a fast DSL connection we ran into a problem with unreliable detections, as there was some 'noise' on the line which caused some package drops. While the standard Internet functions like DNS, HTTP or POP3/SMTP worked well (as they were able to

deal better with connection issues), the 'in-the-cloud' product was not able to deal with such problems. The detection dropped almost to the level of the local AV databases, without any error message, so users wouldn't have been aware of the problem. Only one 'in-the-cloud' solution was affected by this (and the problem has not been fixed at the time of writing the paper).

- *Network traffic*: The least 'in-the-cloud' traffic is generated by solutions which use DNS queries. In the case of HTTP protocols, the overhead is noticeable even if multiple queries to the cloud are combined. HTTPS traffic might create even more overhead, with different advantages. During the beta testing of some new products we noticed that up to 5KB of traffic is transferred per queried file, so an on-demand scan of many PCs in a network (started at the same time) could easily generate a DDoS attack on the Internet connection or even on the 'in-the-cloud' servers.
- *Product availability*: The AV companies first started to integrate the 'in-the-cloud' services for AV detection in the consumer (retail) products in order to protect this target group in a better way. Surprisingly, in some cases, the additional detection was at first used only by the on-demand scanner of the product and not as part of the email or download scanner which are the main vectors for new infections. The technology is not yet widely deployed in enterprise-class products, especially at email and web gateways which are already using 'in-the-cloud' functionality to query web reputation services (WRS) and URL filter lists, for example. Anyway, more AV vendors have announced that they will use their 'in-the-cloud' servers for all kinds of newly released products.
- *Digital signatures*: While all vendors claim to use digital signatures to back-up the detections (to rule out a spoofed detection), we saw that one 'in-the-cloud' solution used invalid SSL certificates for the communication. So the data was encrypted, but it was easy to decrypt the content and change the response to cause all manner of side effects.
- *Response times (time to update)*: One might think that the response times to new threats with an 'in-the-cloud' product might be significantly better than with traditional updates. For many years, we have been testing AV solutions in order to check how long it takes to protect a customer with signatures against a new malware [6]. We have found that the regular database updates of a good number of AV products are still released earlier than 'in-the-cloud' updates. Besides this, more attacks of similar malware were stopped more effectively with regular AV products (thanks to the help of generic and heuristic detection mechanisms) when compared with 'in-the-cloud' services (as most queries are still full file hashes, but the hashes change with every new variant).
- *False positives 1*: It's not a secret that some AV vendors copy their detections from other AV products. With the use of multi-scanner systems it's an easy task: when enough competitor detections are in place it's a good indication that a file might be malware. The danger here is to automatically add such detections to the cloud, without manually checking them beforehand. Since 'in-the-cloud' services are supposed to deliver instant protection, this is not an unlikely scenario, as every manual check would cost time and may therefore be omitted by the vendors. The increased number of false positives that we have seen during the introduction of the first 'in-the-cloud' products may support this claim. The good point is that false positive detections can be fixed much more quickly than with regular AV database updates. Besides this, much more information about 'known good software' to avoid false positives can be stored on the 'in-the-cloud' servers (usually some 100GB to several terabytes), which is not possible with regular AV software.
- *False positives 2*: Statistical analysis can easily be performed based on the nature of the 'in-the-cloud' queries. For example, identical or very similar files are very suspicious when they appear in numbers within a short amount of time. While this could be a new malware outbreak, it's also possible it's just a new version of a popular product or a patch. In the first few weeks of the 'in-the-cloud' enabled products we often saw that new software was blocked in large numbers, but this situation has improved. It looks like such software is no longer automatically blocked, but the criteria is mainly used to prioritize incoming sample submissions (to check the samples first which are more widespread than others).
- *Detection rates*: During our regular testing, we saw that the detection rates of 'in-the-cloud' enabled products are usually better than those of traditional AV scanners. This means that users do benefit from the additional layer of protection. However, these results can be misinterpreted. As the majority of scanner tests focus mainly on detection rates of the on-demand scanners, the results might give a better impression of the product than is actually the case. It is very important that testers focus not only on one layer of protection, but on all features the product has to offer [7]. As most of the current 'in-the-cloud' products are based on full file hashes, the AV scanners are good at blocking known malware. However, they are not able to block unknown threats and the majority of new malware is built for targeted attacks, so it is unique (per target).
- *On-access vs. on-execution*: The AV companies which integrated the 'in-the-cloud' functionality into their products also extended the on-access scanner accordingly. However, in many cases, the cloud is only queried when a user wants to execute a file (starting a program or a malware), but not when the file is copied or otherwise accessed. This can be especially tricky in networking environments. It is also problematic when libraries (like DLL files) are loaded but not executed, as they are not checked. The same applies to scripts where the interpreter executable is checked, but not the script itself. When executing a file where an 'in-the-cloud' query needs to be sent we have seen that the timeout is too strict so that the file can be executed, even if the query reports 'known malware'. Several 'in-the-cloud' products were affected by these issues.
- *Scan performance*: While many people think that 'in-the-cloud' solutions are much slower (due to the network traffic and latency) when compared with traditional products (which have their database on the local disk only), we found that the 'in-the-cloud' queries don't slow down the system performance that much. In some cases

it's even possible that the scanner is faster now, especially on systems with limited memory (as the data doesn't need to be loaded and processed in memory, occupying several tens or even 100MB, but it can be queried with almost no memory overhead).

The list is based on our current testing experiences, but will probably be extended with some more examples in future. While the list mainly includes negative aspects, we want to emphasise that this is based on our current experiences. The products will mature over time and develop further. Future implementations might use more proactive mechanisms and operate more reliably.

CONCLUSION

We have briefly described the potential of 'in-the-cloud' services but also shown that there may be some issues associated with them. It is a new technology and therefore one can expect some problems within the first stages of deployment. However, some limitations just lie in the nature of the approach and need to be considered when implementing 'in-the-cloud' services in the products. Vendors should look carefully at the aforementioned theoretical and practical aspects in order to avoid similar problems in future implementations.

The 'in-the-cloud' services offer an additional layer of protection, but they are not the only protection mechanism of today's AV products. A single technology cannot be the Holy Grail of the entire malware problem; only the combination of different approaches can help to protect the user. When carefully combining the different static and dynamic, local and remote technologies, 'in-the-cloud' services will be a valuable addition in protection. In order to do this, it is important that not only are file hashes used, but that the data of the threats to be analysed is chosen intelligently.

Testers should also take care not only to review the on-demand detection score results (with or without using 'in-the-cloud' technology), but to focus on the whole product. The question which should be answered is whether the user is protected against a threat – regardless of whether it is blocked by a URL filter, the traditional file scanner or behaviour-based (dynamic) protection. It doesn't matter how many viruses an on-demand scanner is able to detect if other protection mechanisms might be able to block current and future threats more easily.

REFERENCES

[1] McAfee. Technical Brief: McAfee Artemis Technology. http://www.mcafee.com/us/local_content/technical_briefs/technical_brief_artemis.zip.

[2] Ször, P. Attacking the Cloud, CARO Workshop 2009, Budapest.

[3] Cole, D. Cloud AntiVirus Forecast: Foggy, with a Chance of Irrelevance. Blogpost at http://community.norton.com/t5/Norton-Protection-Blog/Cloud-AntiVirus-Forecast-Foggy-with-a-Chance-of-Irrelevance/ba-p/93837.

[4] Morgenstern, M.; Marx A. Insecurity in security software. Proceedings of the 15th Virus Bulletin International Conference, 2005. http://www.av-test.org/down/papers/2005-10_vb_2005.zip.

[5] AMTSO: Best Practices for Testing In-The-Cloud Security Products. http://www.amtso.org/uploads/amtso-best-practices-for-testing-in-the-cloud-security-products.pdf.

[6] Marx, A. Antivirus outbreak response testing and impact. Proceedings of the 14th Virus Bulletin International Conference, 2004. http://www.av-test.org/down/papers/2004-09_vb_2004.zip.

[7] Marx A. Malware vs. anti-malware: (how) can we still survive? Virus Bulletin February 2008 p.2. http://www.virusbtn.com/virusbulletin/archive/2008/02/vb200802-comment.