Android Security Product Testing

Maik Morgenstern (CTO), Andreas Marx (CEO)
AV-TEST GmbH, Klewitzstr. 7, 39112 Magdeburg, Germany
Tel +49 391 6075460

Email mmorgen@av-test.de and amarx@av-test.de

Abstract

Asia is the world's most important market for up-to-date smartphone technology with an enormous growth over the last few years. However, mobile malware and hacker attacks have evolved rapidly, too.

In the Android anti-virus world, more than 100 products promise to offer to protect the digital mobile world. In order to make good usage and buying decisions, meaningful test results are required to make the correct personal choice of the anti-malware solution.

In contrast to desktop anti-malware, new security features and a different kind of performance impact have to be considered.

This presentation will describe the AV-TEST approach to test and certify mobile anti-virus and other security products.

Introduction

While there are over a dozen different mobile operating systems, there are only very few that are relevant. As you see from figures 1 and 2 these are primarily Android and iOS. Analysts also expect Windows Phone to play an important role in future, as depicted in figure 2.

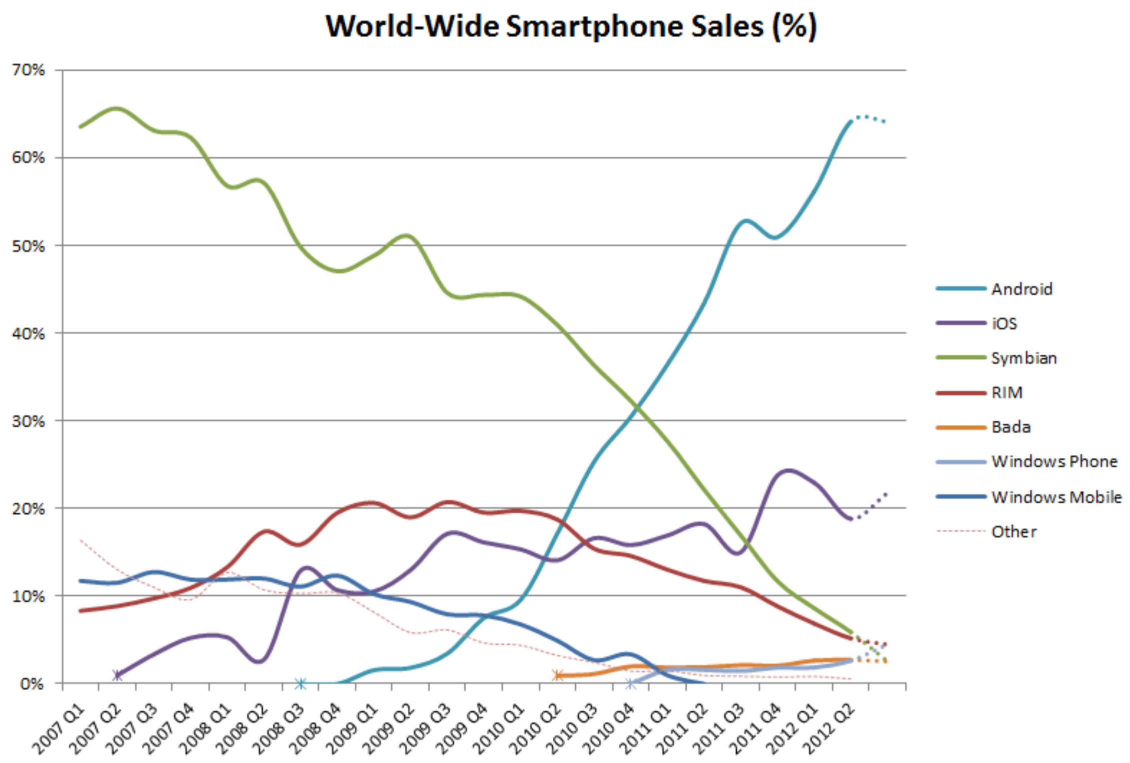

Figure 1: Source http://en.wikipedia.org/wiki/Smartphone#Market_share

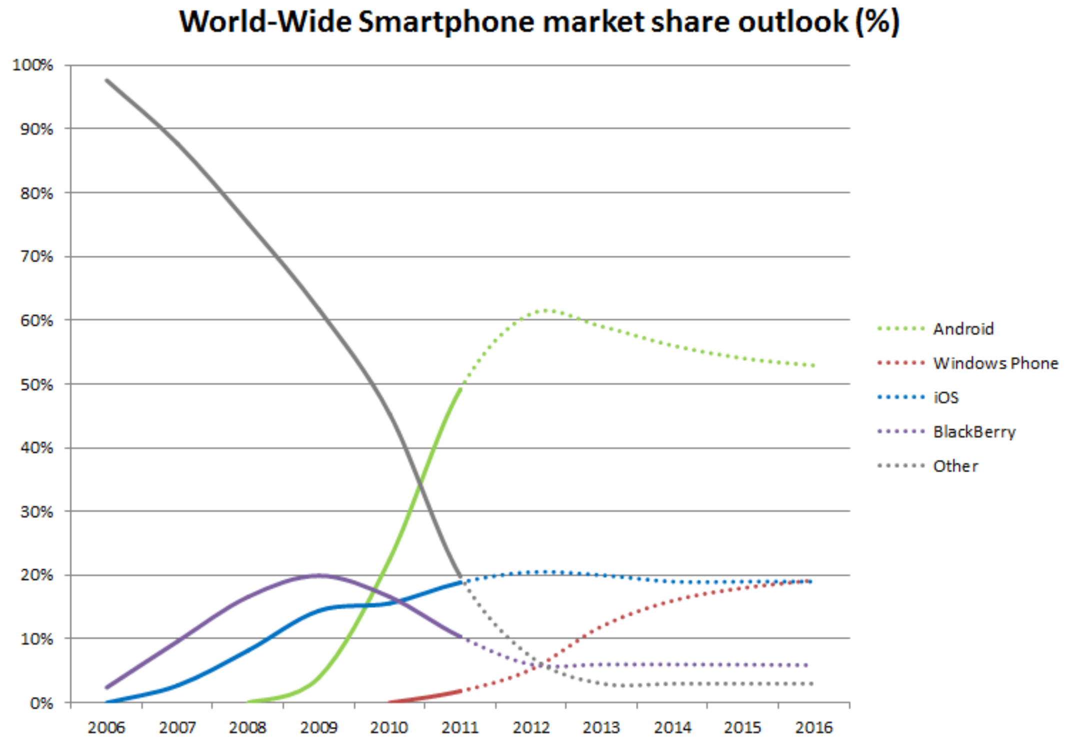

Figure 2 Source http://en.wikipedia.org/wiki/Mobile_operating_system#Current_market_share_and_outlook

One question naturally arises: Why is Android so popular? The answer is pretty simple: It is cheap and easy to use for mobile device vendors, furthermore it is very open and customizable. The openness is also a reason for the success among the users. It encourages developers to create new apps and there are many market places where these apps can be shared and sold. And there are many different vendors and therefore devices of which the user can choose of.

In contrast to this approach there is iOS which is a closed system. There is only one vendor, Apple, which provides a very limited set of devices (iPhone and iPad). There is only one official market, the App Store which is sometimes seen as pretty restrictive. Not every kind of app is allowed in there.

Windows Phone 8 will be somewhere in the middle between these two approaches. There will be several vendors licensing the OS, but the official market will be pretty restrictive, according to announcements that have been made by Microsoft so far.

These different approaches to mobile platforms also result in different security problems on these platforms. While there is no malware problem on iOS, there are already thousands of malicious apps for Android. Since Windows Phone 8 has not yet been released we can only guess what future will bring, but since it seems to be a more restrictive environment than Android, we guess that there won't be a malware problem too soon, if at all. However, malware is not the only security problem. In fact, especially on mobile devices, there is a lot more to security. Besides Exploits (which could be counted as malware to some extent), that could possibly work on all systems, users fear to give up their privacy, lose their data or even the whole device.

So which of the possible security problems are really relevant? Which solutions are there? And most importantly: How can tests tell you, what the right solution is for you?

Since Android is the environment which is the most open one we will look at this operating system and show you the current state of Android security software testing, what the associated problems are and propose a better way to test.

Current Testing and the problems

There have been comparative reviews of mobile security solutions from three testing organizations so far:

- AV-TEST
    - "Are free Android virus scanners any good?" November 2011 [1]
    - "Test Report: Anti-Malware solutions for Android" March 2012 [2]
- West Coast Labs
    - "Custom Test Report - NQ Mobile Inc." October 2011 [3]
    - "Mobile Security Technology Report“ October 2011 [4]
- AV-Comparatives
    - "Mobile Review" August 2010 [5]
    - "Mobile Review" August 2011 [6]
    - "Mobile Security Review" September 2012 [7]

All of these reports have strong and weak points. In order to understand the problems of each report, we will give a short summary and overview of them. None of it is perfect and there will probably be never the perfect test, but looking at the details of these tests will help to build better tests.

AV-TEST, "Are free Android virus scanners any good?"

This test looked at seven free, less known, Android security solutions. It furthermore tested two well known commercial vendors for reference. The products have been tested both with an on-demand scan as well as with a small scale on-installation test. The purpose of this test was to show that there free Android virus scanners that do effectively nothing to protect the user. This was the only statement that the report made: Be careful when choosing a security solution, some may not protect you at all and only display advertisements. The report was not meant to be a comparative review and therefore didn't provide any information which are the best products. But since this wasn't the purpose of the report it can't be blamed for that.

AV-TEST, "Test Report: Anti-Malware solutions for Android"

This report determined the detection rate of 41 Android security solutions against a set of recent Android malware (618 samples, not older than six months). It lists the total relative detection rate as well as the detection rate per family, allowing the user to check the details in regard to different types of malware families. Some families are more PUA than malware, so it may be OK to some users if these families are not detected. Besides the malware detection rate no other tests have been carried out.

WestCoast Labs, "Custom Test Report - NQ Mobile Inc."

A commissioned test (by NetQin) to compare eight mobile security solutions. The test consisted of determining malware detection rates (including a small scale false positive test) as well as checking Anti-Loss and Remote-Wipe features. The malware detection test was carried out on two sample sets. One test set consisted of 75 samples and another one of 129 samples. No information about the age of the samples is given. It also seems that the test set consist of samples for several mobile OS while the test has been performed on Android. The main malware detection test was carried out as an on-demand scan and only a small subset of samples have been tested with the real-time protection. Since some products only scan APKs on installation, they won't detect anything during an on-demand scan resp. they don't offer an on-demand scan at all. So the results will be misleading if you don't read the testing methodology properly (the summarized report doesn't even contain the information about the real-time scanning part).

WestCoast Labs, "Mobile Security Technology Report"

A certification test of three products. The malware detection has been determined, however no information about the test set (age and size) is given. No version information for the products are given either. There are a few words about the other features of the solutions but it does not look like deep tests have been carried out on them.

AV-Comparatives, "Mobile Review"

Most of the following is true for both "Mobile Review" reports from 2010 and 2011. The first report was for products on Symbian and Windows Mobile while the second report covered Android versions. The focus was on features aside from the malware protection. Nevertheless small scale detection test have been performed. There is some information about the sample set in the second report, the Android malware samples from the extended Wildlist (July 2011) have been used. The Wildlist serves as tester's reference, providing samples that are several months old and are supposed to be detected by every vendor. Usually this is the case, since these samples are simply added to signature databases without checking to make sure the product scores 100% in tester reviews. Not surprisingly most products scored 100%. The first report tested four products and the second report tested nine products. Both reports focus on the description of the solutions and their features instead of actually providing test results. There are huge feature lists but no actual test results besides some basic feature descriptions. No false positive or performance testing has been carried out.

AV-Comparatives, "Mobile Security Review"

This is the latest of all reports and covers thirteen products, but still missing some major players like NQMobile, Symantec or AVG. The tested features include malware detection rates, battery usage, false positive testing and reviews of the further features. The malware test set consisted of 18.021 files found between March and July 2012, split in about 75 families. No further information is given, e.g. which families are included and how are the samples distributed among these families. The test has been carried out first on-demand and then on-installation to make sure all products are adequately covered, independently from their protection approach. The false positive set consisted of 200 samples without advertisements. All products detected over 93% of the samples and not a single false positive occurred. The battery test also didn't find any differences among the products. The impact of the products was always less than 3%. The test of the further features is similar to the earlier AV-Comparatives reviews, these are more feature listings and descriptions of them. Some bugs and inconsistencies are reported but no real comparative review has been performed on them.

With that overview it is now possible to summarize some of the problems of all the tests:

- Limited testing criteria
  - Whenever there is a detection test, there should be a false positive test as well
  - There is more to test than just malware detection
    - Performance, Anti-Theft, Encryption, Backup etc. are important as well
    - PUA vs. false positive vs. malware problem
  - Wrong focus
    - Is malware detection the most important feature?
- Bad testing methodology
  - When testing malware detection, it should be feature independent
    - Some products don't provide an on-demand scan, but still protect the user with
  - Results are meaningless when they are all the same for all products
  - No comparative testing of certain features, instead describing the feature
    - Can these features be tested comparatively at all?
- Bad documentation of the test
  - Missing version information or product details
  - Unclear methodology
  - Unclear sample set
- Bad sample set
  - Sample set is too small
  - Samples are too old or not prevalent or both
  - No detailed information about the sample set is given
    - Sample set may be distorted
      - 1000 samples split into 10 families with 100 samples each

- 1000 samples split into 10 families, where one family has 950 samples and the other 50 samples are distributed among the remaining 9 families
- If a product detects that one family particularly well it will score good

- Wrong product selection
  - Important products are missing
    - The results of the products cannot be realistically rated, as the winner of the test might still be worse than the missing product
  - Too few products tested
    - If the product the user is looking for/using is missing, then the test doesn’t help him
  - Comparing apples to oranges
    - A mediocre product will always look good compared to a fake product
    - Different products have different features
- Bad Timing
  - Tests are quickly outdated
    - Threat Landscape and overall mobile landscape changes fast and often
    - Lots of new malware samples evolve daily
    - So the products change fast and often as well
  - Therefore up-to-date tests are necessary as well as regular tests to show the development of the products and to make sure the vendors are able to consistently react to the new problems
- Results don’t help the user to choose the right product
  - Do these tests answer the questions of the user at all?
    - All products achieve more or less the same results
    - There are no comparable results at all, just plain descriptions
    - There are testing criteria missing, only certain features/scenarios are tested
    - There are too many plain results, interpreting them is hard to impossible for normal user

With all these problems in mind it is necessary to look at the issue from the other side which we will do in the next section.

What really Matters

Now that we know what the problems are, let’s try to ask the right questions and try to find a way to answer them.

When designing tests it is necessary to look at the products from a user’s point of view and then try to abstract their requirements to come up with repeatable, comparable and verifiable tests. Below are a few questions that we think many users have when thinking about mobile security and when looking for software to help them. We also list some of the features or other factors that may be important for test design:

- What happens when I lose my phone?
  - Can I get it back?
    - Anti-Theft (Locate Device)

    - Is my data safe?
        - Remote Wipe
        - Remote Lock
        - Encryption
    - Can I get my data back?
        - Online Backup
- Is my privacy ensured?
    - Which apps spy on me and can security software tell me and protect me?
- Is malware a problem for me?
    - Malware Detection rates
    - I like free games, but they are reported as bad
        - PUA vs. malware
- I want to protect my child from inappropriate content on the phone.
    - Parental Control
- Will the security app eat all my battery or download bandwidth?
    - Measuring impact on battery life and downloads
- Where do I find recent test results of my product or of all the good important products?
    - Required to perform regular tests to always have up-to-date results of the recent product versions
    - Include as many products as possible
- Has this product always been so good/bad?
    - Again regular tests, so the user could look through the history and watch the development of a product.

These are quite a few questions and requirements. It is certainly nearly impossible to find the perfect test to answer all of them. But we will still try to propose a test design, that AV-TEST is going to implement. See the next section for details.

How to solve the problem

We are not going to talk about too many technical details here. This has already been done by Hendrik Pilz, Director of the Technical Lab at AV-TEST, on Virus Bulletin this year [8]. See his great presentation for those details. Also you can have a look at the AMTSO papers, while not specifically written for the mobile world, contain a lot of good points that should be kept in mind [9].

We will talk about the more general testing design.

One obvious approach to solve the problem is quantity:

- Test as many products as possible
- Test as many aspects of the products as possible
- Test as many scenarios/samples as possible
- Perform the tests as often as possible

That would at least provide all the required details to answer the above questions. But no average user would be able to dig through all these raw data and the question remains whether everything can be adequately tested at all?

So we will have to refine that approach step by step. Let's start with some general parameters.

Timing

In order to achieve regular and up-to-date results we will perform a test on every second month. An even smaller timeframe would be hard to achieve, since the tests alone run a few weeks, then there is the verification and the publication of the results. With that we will have six results in one year, it will be never older than two months and it will be possible to cover new product versions very quickly. This ensures that users will always find up-to-date results and can examine the development over time.

Product Selection

Since resources are usually limited it is not possible to test all products that exist. So we decided to include at least all important vendors. We will most likely end up with 20-30 products here and are confident that these will cover the products most users are looking for.

Basic Information

Detailed version information about the tested products should be given, so that users can identify the correct product version. Also clear information about the used samples and testing methodology has to be provided.

Now that these general parameters are defined we can go on to the more interesting part of the test design. What are the individual aspects/scenarios that should be tested? These are at least the following ones:

- Malware detection rates
    - Including PUA (e.g. aggressive adware)
- False positive rates
- Performance impact
    - Battery drainage
    - Download bandwidth
- Further security features

The testability of these aspects differs a lot, so we will go through these one by one.

Malware detection rates

There are two things to define here. The testing methodology and the creation of the sample set. The testing methodology has been explained in detail as outlined in [8], so we will only give a very brief overview. Since products implement different features, we have to find a feature independent way of testing. Not all products are capable of performing an on-demand scan. But all products should be able to protect against malicious apps once they enter the system. Therefore we test exactly that: We try to install a malicious app on the phone and check whether it is blocked or not. If the product does have on-demand scan functionality, then we will use that to reduce the number of samples that have to be tested with the installation approach. In terms of testability we encourage all vendors to implement an on-demand scanning function, since this enables much better and faster tests.

The second important point is the creation of the sample set. It has to consist of up-to-date and prevalent samples, since malicious apps are usually quickly removed from official market stores. Only really fresh samples can provide meaningful results. Testing against old samples, that are known to the vendors for months and that are not seen in the wild anymore, doesn't generate much useful information. We will only test samples that are not older than four weeks before the start of the test, e.g. if the test starts on 01.01.2013 then the oldest sample in this test will be from 01.12.2012. Furthermore we are going to remove duplicate samples to prevent distortion of the results. We will end up with around 1000 samples then, which is a good trade-off between testability and relevance of the sample set. There are many Fakeinstallers with different hashes but the same APK name. While these samples can be treated as different, due to the different hash, they are only very slight variations of each other. This becomes important when thinking about the distribution of samples among the different families. It is easily possible to fill up a sample set with thousands of Fakeinstallers and if one program detects this specific family it will detect all of its variations and if it doesn't detect the specific family it will miss all the slight variations:

	Samples	Product A	Product B
Fakeinstallers	900	900	0
Other families	100	0	100
Final Result	1000	90%	10%

Product A detected only the Fakeinstallers samples but missed out on all other families, while Product B only missed the Fakeinstallers family and detected all other samples of all other families. Due to the heavily distorted test set, Product A would score a very good overall 90% while Product B only scores 10%. In reality Product B would protect against more threats while Product A would be useless against everything but Fakeinstallers. Therefore it is important to have a realistic distribution of samples among families. It is OK to have more samples in one family than in others, e.g. if one family has more variations or is more prevalent and more samples are in the wild. In order to understand the numbers, the details of samples per family should be published together with the results.

Another important aspect of sample set creation is the question which samples should actually be included in the set? There are at least three different groups of malicious or unwanted apps on Android:

- "Real" malware
- Commercial spying/hacking etc. tools
- (Aggressive) adware
- Privacy Threats

Definition of malware and the other possibly unwanted apps differ a lot among the different vendors. Nevertheless we are going to have at least two separate sets to test for detection rates. The first will be the malware test set which only contains real malware. The second set will contain anything else that would be called PUA (all the three remaining sets from above). Depending on how the vendors and users will see this in future we might have to separate the PUA set even more.

False Positive Rates

The testing methodology is pretty much the same as malware testing. To make sure that all products are really tested, the clean samples should be tested on-installation. Selection of the actual samples is more interesting, because of the PUA problem we just pointed out above.

There are two interesting types of apps for false positive testing:

- Apps that are widespread among users
- New versions of popular apps

Therefore we decided to create our test set out of these two criteria, choosing 100 apps from the Google Play Top Lists as well as 100 from the Google Play Top New lists. But the sample selection doesn't stop here. Even when the app comes from the official Google Play store and is not malware, there may be reasons why it will be flagged by security apps. So we have to carefully remove apps that contain (aggressive) adware, that are a threat to your privacy or that can be considered spying/hacking tools. This can be very difficult, since nearly all free apps contain advertisements and many popular apps can be considered a threat to your privacy (remember Facebook). Therefore it is absolutely necessary to verify results after the test and list the actual false positives together with the test results, to help the user make their decision.

Performance Impact

We already know that there are two big concerns when thinking about performance impact on the mobile device, battery drainage and download bandwidth. To test battery drainage there are pretty much two approaches. Either you are measuring it directly or you are measuring it indirectly (by measuring the used CPU cycles). The first option clearly gives a direct result but it is impossible to perform this test right. You will have to render out all external influence factors:

- Temperate of the environment has to be constant as well as of the device and especially the battery
- The 3G/WiFi signal has to be exactly constant
- The behavior of the device has to be exactly constant
- The condition of the battery has to be exactly constant

It is impossible to guarantee these conditions for all products in all test runs, since there are too many variables to control that change all the time, even in a lab environment. Therefore we opted for the second choice and measured the CPU cycles that are used by the product when performing a defined set of actions on the device. We verified our approach by comparing it to the first approach (a simplified one without taking care of all external factors!) that showed the clear relation between used CPU cycles and battery drain. The two following charts illustrate that relation:

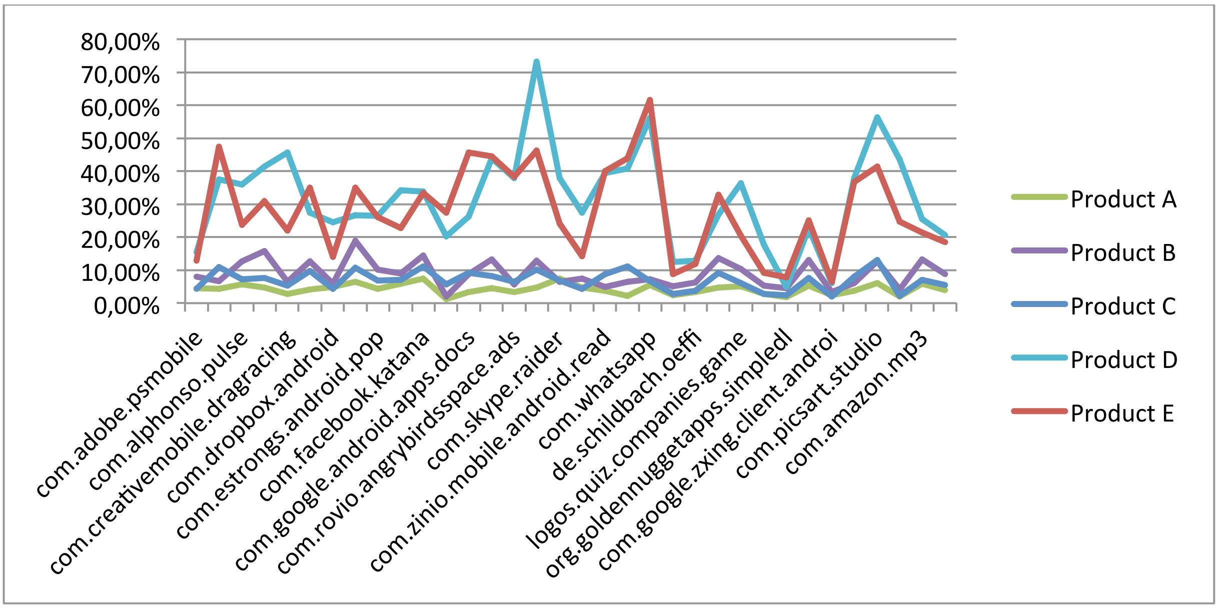

Figure 3: Average CPU usage chart during the installation of 35 apps from Google Play

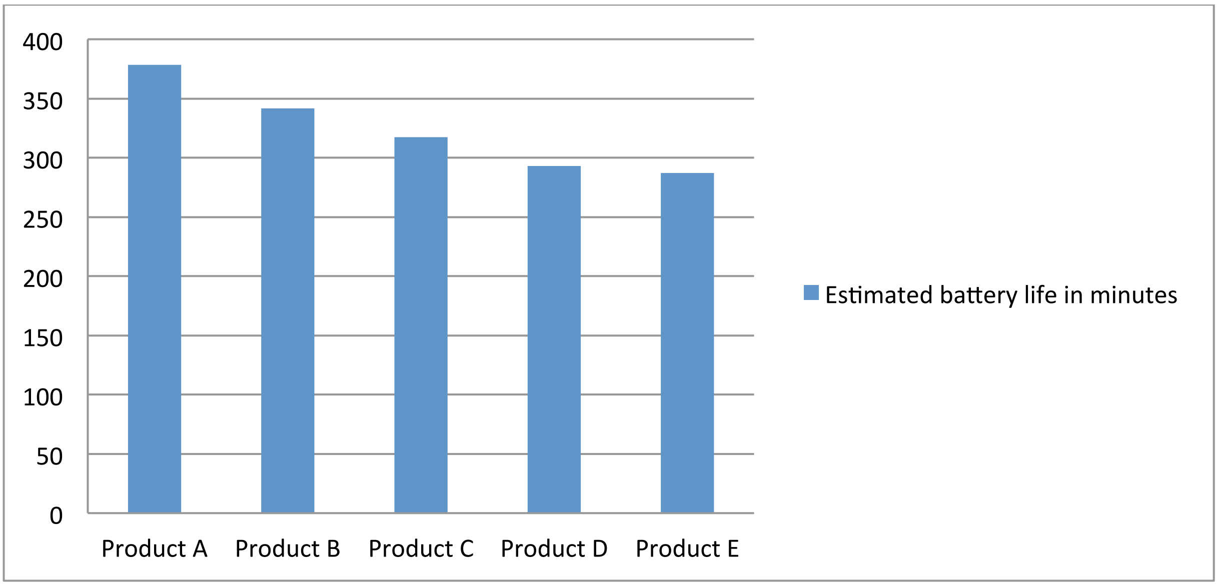

Figure 4: How long would the battery live when the test runs indefinitely?

Products D and E require the most CPU cycles while performing the test tasks and consequently the estimated battery life would be lower than for the other products.

The set of actions that should be run to measure the performance impact should be chosen carefully as well. We will perform at least two different sets. One will replicate a normal user's behavior and contain the following actions:

- Sending Receiving Messages (SMS, Mails)
- Making Phone Calls
- Browsing the Web
- Downloading/Uploading Pictures
- Watching Pictures/Videos

These are all actions that shouldn't bother the security app too much, so we don't expect many differences among the apps here. Probably we won't be able to measure any significant differences at all unless there is a bad implementation in one product. So unfortunately this real-world test will not much meaning most of the time.

In order to support the developers of security apps we also implement another set of actions which will show more differences among products. This is the installation of apps on the phone. Since all the security apps perform their checks during the installation of new apps we can be sure that they will work quite heavily here. The results of these tests have already been depicted in the charts above.

As far as download bandwidth is concerned, the same two sets of actions can be used and the download bandwidth can be measured on the phone.

Further Security Features

The aforementioned testing criteria were all testable in a comparable and reproducible way. The further security features are usually less testable. They differ a lot in their actual implementation and features. They do different things and they do them differently. Not all products have the same feature set or mean the same thing with the same name. That is the reason why there haven't been any real comparative reviews of these further security features.

The only possibility to test these is again to be as feature independent as possible and try to look at real-world scenarios. Let's take the following example:

1. Phone is lost
    a. What can be done to protect the data?
    b. What can be done to get the data on another phone?

There would be at least three ways how security software could help in case of 1.a:

- If it offers an encryption feature, all the data is encrypted and can't be accessed by the thief
- If it offers a remote lock feature, the phone can be locked so the thief can't access the phone
- If it offers a remote wipe feature, the data can be wiped so the thief can't access the data

If a product does offer any of the three features it would be necessary to test how well this actually works. This can only be done in a very basic and straightforward way (Does it work? Yes/No) since there are dozen approaches to test the single features and what-ifs. The same can be done for other real-world scenarios.

To summarize all of the above:

- We have to perform plenty tests (every two months)
- We have to include plenty products in the test
- We have to look at plenty test criteria
- We have plenty data in the test reports

This is a lot of work and those big data piles don't help any average user to find the best product for him. It is necessary to work a bit more on the data and transform them into easier messages. One simple message is a certificate/seal which will be awarded to products meeting certain criteria. So the user knows all products that wear this seal are OK to use.

But there are different user groups and they may have different requirements. In order to help them it may be useful to look at their usually behavior and their requirements to find features that are especially important to them and features that may not be as important. Sometimes performance issues are not as important malware detection and sometimes encryption is essential while parental control might be more important for others. When keeping this in mind it becomes clear that different products could be suitable for different user groups. Ideally one could define a few user groups with their different requirements and the best fitting products, so that every user would find its user group and the according product. Something similar could be done for certain scenarios (or combinations of them) instead of user groups.

Conclusion

The paper shows that so far there has not been a single report on Android security apps that was the perfect help for any user. They all had serious downsides, and are outdated by now anyways. We looked at the strong and weak points of the reports and found quite a few challenges when trying to come up with better reports. The key point is to answer the questions user really have. In order to do this right, quite some effort has to be spent. A lot of products have to be tested, on a regular basis with several test cases and scenarios and the results have to be interpreted so that the normal user will understand them.

No all of this will be working in one single test (or a series of tests). Different test labs will have different point of views to the same question and may give different answers. While we hope that our regular tests are a big step forward, we are sure that this is not the end of the development. By end of next year we will look totally different at these tests and many changes will have been implemented by then.

References

[1] AV-TEST, Are free Android virus scanners any good, November 2011, http://www.AV-TEST.org/fileadmin/pdf/avtest_2011-11_free_android_virus_scanner_english.pdf

[2] AV-TEST, Anti-Malware solutions for Android, March 2012, http://www.AV-TEST.org/fileadmin/pdf/avtest_2012-02_android_anti-malware_report_english.pdf

[3] West Coast Labs, Custom Test Report - NQ Mobile Inc., October 2011, http://westcoastlabs.com/downloads/productTestReport_0070/NetQin_Custom_Test_Test_Report.pdf

[4] West Coast Labs, Mobile Security Solutions, October 2011, http://westcoastlabs.com/downloads/techReport_31/Mobile_Security_Solutions.pdf

[5] AV-Comparatives, Mobile-Review August 2010, August 2010, http://av-comparatives.org/images/docs/avc_mob_201008_en.pdf

[6] AV-Comparatives, Mobile-Review August 2011, August 2011, http://av-comparatives.org/images/docs/avc_mob_201108_en.pdf

[7] AV-Comparatives: Mobile Security Review September 2012, September 2012, http://av-comparatives.org/images/docs/avc_mob_201209_en.pdf

[8] Hendrik Pilz, Building a test environment for Android anti-malware tests, Virus Bulletin Conference 2012, Dallas

[9] AMTSO: Documents and Principles, http://www.amtso.org/documents.html