Guidelines on Mobile Testing

Notice and Disclaimer of Liability Concerning the Use of AMTSO Documents

This document is published with the understanding that AMTSO members are supplying this information for general educational purposes only. No professional engineering or any other professional services or advice is being offered hereby. Therefore, you must use your own skill and judgment when reviewing this document and not solely rely on the information provided herein.

AMTSO believes that the information in this document is accurate as of the date of publication although it has not verified its accuracy or determined if there are any errors. Further, such information is subject to change without notice and AMTSO is under no obligation to provide any updates or corrections.

You understand and agree that this document is provided to you exclusively on an as-is basis without any representations or warranties of any kind whether express, implied or statutory. Without limiting the foregoing, AMTSO expressly disclaims all warranties of merchantability, non-infringement, continuous operation, completeness, quality, accuracy and fitness for a particular purpose.

In no event shall AMTSO be liable for any damages or losses of any kind (including, without limitation, any lost profits, lost data or business interruption) arising directly or indirectly out of any use of this document including, without limitation, any direct, indirect, special, incidental, consequential, exemplary and punitive damages regardless of whether any person or entity was advised of the possibility of such damages.

This document is protected by AMTSO's intellectual property rights and may be additionally protected by the intellectual property rights of others.

AMTSO Guidelines on Mobile Testing

Executive Summary

The purpose of this document is to explore the important issues that exist when testing anti-malware software on different mobile devices.

While the document focuses on mobile platforms it is important to understand that little of this overrides the previously-published AMTSO principles of good testing. As such, this document should be read in the context of these other documents. For example, correct sample selection is just as important in the mobile space as it is on the PC platform.

Perhaps the most important differences between PC and mobile testing relate to the extra constraints to which mobile devices are subject. These include limited user privileges, bandwidth and power.

With power consumption it is important that the real nuances of power drain are measured properly during a test. For example, a significant contributor to draining power from a phone is its screen and radio. The amount of power used can vary due to time of day (auto-dimming of the screen at night) and network availability (the phone may try and boost its signal under certain circumstances). Both of these effects (and others) can cause significant differences in battery usage.

Many end users have phone plans that incur charges as they are used and so bandwidth use is of great interest. Furthermore, for those products that leverage cloud-based capabilities, connectivity may have an impact on a product's abilities to protect the device.

Finally, the configuration of the phone is critical and must be described completely. A good example of how important this can be is whether the phone is "rooted" or not. A rooted phone allows both the malware author and the anti-malware software more access and can, in principle, change the results of a test. Such configuration details need to be provided to the reader so they can read the test in context.

Ultimately the recommendations are that testers document test in detail, so that another person could attempt to replicate the results presented.

Introduction

This document covers issues that are specific to the testing of malware and security countermeasures on mobile devices, such that tests are meaningful and repeatable. The document outlines additional issues involved in best practice testing, above and beyond other AMTSO guidelines and best practices. This document is not a comprehensive listing of all such issues.

Unless otherwise defined herein, all terms included in this document are used with their common meaning. This paper uses the term 'mobile device' to refer to Android-based smart phones, tablets and other personal devices.

AMTSO documents are best read in conjunction with the *Fundamental Principles of Testing* and other documents on the AMTSO documents page at www.amtso.org.

Scope

This document is designed to focus on the testing of anti-malware solutions on the Android platform. We have chosen to focus on Android because of its prevalence, rapid adoption, the availability of antimalware solutions and the fact that it is being actively attacked by "real world" threats.

The comments made here are primarily applicable to mobile devices though in principal many of the recommendations will apply to any Android based system.

Similarities with PC Platforms

In many ways, the testing concepts that have been well-developed on the PC platform may be used when testing mobile devices. Android phones are essentially Linux-based personal computers that can provide similar levels of computing power to that available with reasonably modern desktop PCs. For example, the Google Nexus 4 smartphone has a 1.5GHz quad-core processor and 2GB RAM, while it was still possible at the time of writing to buy a similarly-specified laptop PC (albeit with an Intel/AMD processor).

One major difference between PCs and Android (and Apple iOS) devices is that the latter mobile devices restrict the user's control over the system. However, it is possible to gain full access by 'rooting' the device, which bypasses the inbuilt protection and bestows administrator-level (or 'root') privileges upon the user. While rooting increases the possibilities for monitoring the system, there are less positive issues involved when testing security products. For more information, see the section entitled "Rooting and Elevated Privileges" in this document.

Real World Anti-Malware Testing

Malware Samples

It is preferable to test in a similar way to the experiences that users have in the real world. This means using a selection of malware samples that are coming into contact with users.

Products should be configured in a realistic way and third-party apps usually found on mobile devices should be pre-installed on emulators if used. Factory default settings for physical mobile devices (as opposed to virtual machines) are appropriate.

Malware should be introduced to the target in a realistic way, from appropriate third-party app stores for example.

Emulators

Emulators provide a wide range of possibilities for monitoring and manipulating the installed software (including malware). However, they are also limited by a number of factors. These include a lack of true

GSM connectivity, which precludes the ability to connect the emulated system to real voice call or SMS services.

Other features, such as Bluetooth and degrading battery charge level and charging states, were unavailable at the time of writing. Battery levels can be set, but won't expire. For more on testing mobile battery use see the section entitled "Battery Drain Measurement" in this Document.

Emulators for malware testing are not sufficient even for legitimate software (FP) testing as not all software will be fully functional in an emulated environment. For example, malware may check for mobile device details that are missing from emulated devices. The malicious application may not install as a result.

Emulators are very slow to run, even when run on powerful PC hardware.

Battery Charge

Some anti-malware products disable themselves when the device's power runs low. Testers should monitor this situation and note if and when an anti-malware product stops working effectively. One option might be to introduce the EICAR file or some other standard detectable file repeatedly. Other features may also take priority. Some apps may become more aggressive when external power is supplied, which could influence performance impact testing. A product may scan more thoroughly or create backup files. This could affect data and processor use.

One solution is to ensure that the phone in continually charged. Another is to note that in real life phones do run out of power and that some products will cease to protect them under specific circumstances.

Testers can override the device's own power management system and tests can be conducted at different (simulated) levels of battery charge. See the section entitled "Battery Drain Measurement" in this document for more information.

Pre-Installed Software

Some mobile devices could come with anti-malware software preinstalled, which means that the security product could be given root privileges. This in turn could change how well this product performs on that platform as opposed to others. It may be more or less effective and this configuration must be noted clearly in the test report and the conclusions should reflect this fact.

Some mobile device versions may be more or less vulnerable, so this too could affect test results. This could also affect sample selection if usable samples are required to test an anti-malware product's full protection capabilities.

Testers are advised to compare products with similar privileges. If this is not possible or desired then the conclusions of the test should make clear that certain (pre-installed) products enjoyed the advantages of having higher privileges than the other (retro-installed) anti-malware products.

Note also that pre-installed anti-malware products may be outdated. Newer versions may be available from app stores.

Connectivity

Mobile devices have different levels of connectivity via 3G, 4G, Wi-Fi and slower technologies. This can affect the features and functions of security software and threats.

Anti-malware products on desktop platforms often perform frequent queries to backend servers and almost all require regular updates. The same is at least partially true for today's mobile anti-malware products. Efficacy may suffer at slower speeds, as some products may either fail in their attempts to communicate with the vendor's systems or may elect not to use up all of the bandwidth in order to improve a user's overall experience.

Location and time can affect the speed and power consumption of devices due to network congestion. Weather and atmospheric conditions can also cause measurable differences and should be considered.

For more information, see the section entitled "Bandwidth" in this document.

Mobile Device Manufacturers

Different mobile device manufacturers and/or carriers frequently have their own sets of configurations for the Android operating system different from the standard build. Some of these settings can make the device more or less vulnerable and thus affect the test results and conclusions. Compared with PCs, differences in hardware for mobile are more profound so there's more need for specificity and scope during tests.

Ideally, it is recommended to use the most common model of a mobile device with the most deployed firmware image of Android. A good choice would be a popular model that is available with a firmware that has not been modified by a mobile network operator/carrier.

For example, at the time of writing the Google Nexus 4 phone was popular, inexpensive and lacking any third-party apps or modifications made by a mobile network operator. It follows that pre-installed anti-malware may not be ideal, as it is (currently) the result of a carrier modification.

Testers may consider creating and deploying their own Android images in the same way that some large business do. This would permit a completely consistent and controlled testing environment, in much the same way as testers install relatively clean installations of Windows when testing desktop PC anti-malware products. However, custom Android installations are not common in the real world so those who create them for tests should document thoroughly all steps taken.

Exact version information should be provided for the firmware and hardware used in the test. In mid2013 there were three dominant versions of Android in popular use:

- Gingerbread (2.3.3 -2.3.7) - 39.7%
- Ice Cream Sandwich(4.0.3 -4.0.4) -29.3%

- Jelly Bean (4.1.x)- 23.0%

(Source: http://developer.android.com)

Common Testing Issues

There are many issues that could have a significant impact on real world testing. The following list highlights some areas that testers should pay particular attention to. As with any real-world testing, it is critical to ensure that the testing environment and other elements of the test approximate as closely as possible the most commonly-used hardware, software, networks, configurations and other variables.

At the very least, testers should note the following details as they appear in the test environment, even if they choose not to address them.

1. To root or not to root? See the section entitled "Rooting and Elevated Privileges" in this document.
2. Allow/disallow installation of applications from third-party markets. E.g. "Unknown sources" setting "allow installation of apps from sources other than the Play Store".
    a. This affects sample selection of malware and legitimate software (for 'FP' testing) significantly.
    b. Third-party markets are enabled by default in many cases. Testers should explain why they chose to include or exclude such markets.
3. Discover if Trusted Credentials is turned on or off by default.
4. Non-standard third-party applications (e.g. remote configuration software) may be hidden but running in the background, interfering with security software. Non-standard system, application, user interaction and performance-monitoring software may interfere with security products.
    a. Note: Some carriers including AT&T monitor some activity on the phone.
5. Non-standard system configuration may prevent security software from functioning. These include:
    a. SD-card mounting. An APK creates a file and then creates a mount event each time. This event can be monitored by anti-malware software. In this scenario the anti-malware product may impact on system performance as it monitors each mount and dismount.
    b. The use of roaming blocking and other non-standard settings profiles.
6. How messaging, email, browsing and software installation is integrated in the UI. Non-standard apps and configurations may affect the anti-malware product's ability to produce effective alerts. Non-standard configurations may also interfere with security products' functionality as these applications are the most common areas monitored since they are the primary vectors of infection.
7. Missing standard software such as the web browser, messaging apps, the system lock and Google Play store. These situations may arise when providing locked-down profiles such as for children.

8. The presence and configuration of the Google app verification feature ("Verify apps" – "Disallow or warn before installation of apps that may cause harm"). This is present in Android 4.2 and could be disabled at the time of this writing. Note: if it is "on," it may interfere with testing the detection abilities of anti-malware products.

Bandwidth

Most anti-malware products utilize bandwidth to be able to function comprehensively, if not completely. In contrast to PCs, bandwidth can be a scarce resource in mobile devices, especially with users with budget data plans, so it can be useful to compare bandwidth usage of different security products.

As a general observation, anti-malware products should aim to be smart enough to avoid data usage on limited networks. The following areas or features are of interest when planning tests:

1. Cloud lookup technology is here to stay so testing the average bandwidth consumption of each product is likely to be of significant interest.
2. Database updates can have an effect on bandwidth use.
3. Does the product avoid cloud lookup and database update function when the phone is in roaming mode? This information informs a usability study about bandwidth use and is also extremely important when conducting anti-malware protection testing. For example, if live database lookups are avoided because the phone is in roaming mode then this fact needs to be made clear in the report. Unless the real-world test is of anti-malware protection while roaming, it is advisable to disable roaming mode in such circumstances.
4. Check if the product performs database updates only on Wi-Fi connections by default. If so, malware exposures should be made using the Wi-Fi connection or else the report needs to explain the inherent limitation faced by the product/user.
5. How much data does the product send to the backend?

Features in Mobile Security Suites

Android AV products may contain more security features than the traditional anti-malware and potentially unwanted application (PUA) file detections. Such features might include safe web browsing and host intrusion prevention systems (HIPS) protection layers. They may also lack some features commonly found in desktop anti-malware products, such as the ability to remove malicious applications that have been installed on the device.

Security features that may be present, but are not directly related to anti-malware frequently include anti-theft measures. It is possible and valuable to focus on these types of specific features only, such as in a test to determine which products are most effective in locating and/or disabling a stolen or lost phone.

However, when conducting a whole product anti-malware test it is important not only to investigate the file detection capabilities of an anti-malware product but the other protection features the product

offers. These features can work hand-in-hand to bring about a significantly more effective protection experience for users.

Exposing a mobile device to threats using realistic methods can expose different layers of protection. Such methods might include one or more of the following ways in which the user obtains an application:

- Browsing to a certain URL for downloading apps via exploits and/or social engineering techniques.
- Visiting official or third-party app markets that host malicious software that spreads via social engineering techniques.
- 'Side-loading' the app from a removable media such as an SD card.

Furthermore, it would be useful to monitor and record how the security applications handle encounters with malware. The following list contains some possible results:

- Recognizing the app before installation: Detecting and differentiating the malicious/PUA App from legitimate apps.
- Recognizing the running app: Detection of possible suspicious app behaviors during its execution.

Rooting and Elevated Privileges

The question of 'rooting' devices, in which a user gains a level of control over the system that is higher than allowed by a default installation, does not have a clear answer. At the time of writing, it is likely that most consumers are using handsets that have not been rooted. Therefore, a real-world test operating on the principles that the equipment used should be as close to the norm as possible should avoid using rooted phones.

Applications installed by a user (such as anti-malware software) do not have sufficient rights to uninstall/remove another installed program. At this time of this writing, Google could achieve this goal automatically, using its so-called 'kill switch', which is able to remove an application but is not able to clean up files that the malicious application has spread across the system. However, anti-malware programs may lead a user to agree to remove a malicious application – perhaps making that removal more easy than would otherwise be the case.

Some mobile devices could come with an anti-malware product pre-installed. It is generally recommended not to use such mobile devices for testing due to the unique advantage that the preinstalled security product may enjoy due to its superior system privileges compared to products installed by a user. Using a standalone version of the product is preferable in normal situations. However, if it becomes common practice for handset suppliers to pre-install anti-malware products then comparing such handsets with different products seems fair.

If testing an anti-malware product on a rooted device is necessary it follows that the other products due for comparison should be installed on rooted devices too.

Corporate Environments

Corporate mobile devices may be more managed than the consumer counterparts and ought to be configured toward maximum protection. Such devices may have the following 'default' settings imposed by administrators or suppliers, which may have rooted the phone to achieve certain goals:

- Using a carrier-specific or custom-made build of the Android operating system.
- Unknown sources not allowed.
- Anti-malware product may be pre-installed.
- Remote configuration software may be installed.

When constructing a test of products and/or handsets intended exclusively for business use, tests should consider all of the above possibilities and research common customizations made by large businesses to mobile devices.

Battery Drain Measurement

This method is suitable only for a specific subset of phones with removable batteries.

Measuring the battery consumption of mobile devices, especially in the case of smart phones, is not as easy as it looks. Once a strong methodology is in place it will be possible to compare a phone's baseline power consumption without an anti-malware product with its consumption with security software installed.

For measuring battery consumption of mobile devices we recommend the following steps:

1. Ensure a stable testing environment that maintains temperature and other physical variables.
2. Use ISO calibrated devices for measurement.
3. Overrule the phone's power management to avoid influence by lighting conditions and local signal strength. Also standardize volume levels to consistent levels.
4. Use automated tasks/workloads that are repeatable.
5. Results must be measured in high detail, with a resolution of nanowatts.
6. Use a dedicated Wi-Fi network for the test.
7. Use a dedicated UMTS base station for the test.
8. Measure the current lost due to resistance in the cables between the measuring device and the phone.

ISO Calibration

It is very important to use calibrated measurement devices, as the measurement differences could influence the test results. When using non-ISO calibrated devices ensure that the measuring is correct by using a frequency measurement.

Measure the Loss of Cable Connection

To measure the correct battery consumption of a device in high detail, down to a nanowatt level, and to discover any influence on that consumption by an anti-malware program to the device, it is very important to exclude any influence of cable resistance. So it is necessary to determine the power loss caused by the wiring between the device and the measuring equipment.

The following chart demonstrates the basics to get SMU specified performances to understand the wiring requirements when using an appropriate device.

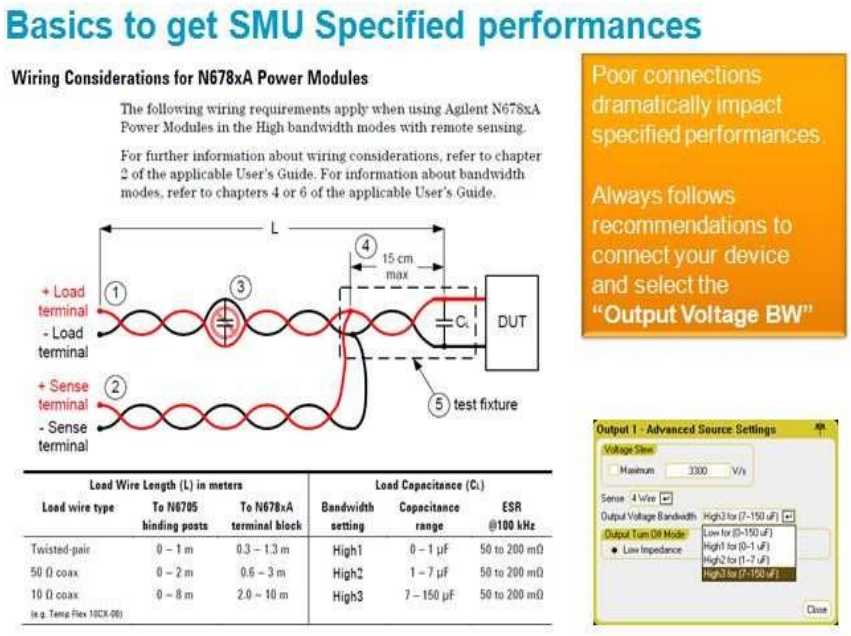

Figure 1: Wiring requirements when connecting a power measurement device to a mobile device (Copyright AV-Comparatives / Agilent Technologies)

Power Management

Sometimes the mobile device will try to adjust the brightness of its screen (or other features) according to the local environment. To avoid this happening, as it will affect the power consumption, the mobile device's automatic power management system should be disabled.

Wi-Fi

Connections to the Wi-Fi base station, and the distance between the mobile device and the base station, influence the battery consumption. Ensure that the distance and the network connections to the Wi-Fi base station remain the same for each test case and, ideally, for all tests that will be compared with each other in the future.

UMTS/3G Base Station

Use a dedicated UMTS base station while testing. Generally-available UMTS/3G networks, as provided by regular mobile operators in a tester's locality, will cause fluctuations in the power measurement results, which may be caused by weather, other connected phones and various other factors.

Simulating the User

To measure battery consumption of an anti-malware product properly, it is essential to simulate the real user. It does not make any sense to measure one single operation, such as a 30-minute phone call, and nothing else. Try to analyze user behavior and simulate this when testing battery consumption.

An example of a user behavior survey is available at www.av-comparatives.org. We recommend that testers perform their own surveys to fit their readers' needs.

Use Measurements of Nanowatt Resolution

In Fig. 2 below you can see the consumption of a smart phone both idle and handling a voice call.

Figure 2: Measuring power consumption requires specialized hardware and software external to the device being tested (Copyright AV Comparatives)

Mistakes to Avoid

Never Use Multi-Meter Breakpoint Measuring

Measuring breakpoints with ampere and volt readings, and calculating the wattage (even if you set it to 100 points a second) does not give accurate results. The energy level of the difference with and without anti-malware software present is too small to be measured with breakpoint measuring.

You cannot synchronize to breakpoints accurately, even if it is two-channeled device. The peaks sometimes only occur in milliseconds so they will not always be measured.

A battery drain analyzer is needed for long-term measurement of true battery consumption. Those measurement devices are extremely expensive; however, there may be opportunities to rent the required equipment. An AMTSO member has indicated that they may be able to help others secure a proper device; for more details contact AV Comparatives (http://www.av-comparatives.org/contact/).

Remove the Battery

Do not measure power consumption with a battery installed inside the phone. This influences the results as the charging status of the battery can vary.

Conclusions

In this paper we have outlined some of the special considerations that are required when testing security software on mobile devices such as smartphones and tablets. These issues are in addition to the typical challenges testers face with respect to false positives and sample selection. In particular, we focus on the role of the network in power consumption and on device configuration.

Perhaps the most important point to note is that the test must adequately document the methodology used for evaluation. In addition to the usual descriptive items, such as criteria for sample selection, mobile devices also require a fairly sophisticated description of device configuration and power drain methodology if power consumption is to be reported.

Finally we note that, as is so often the case, the ability to document the test adequately so it can be analyzed and reproduced is critical to maximizing the test's utility.

This document was adopted by AMTSO on February 20, 2014