MRG Effitas Android AV review 2018 Q4

Contents

Introduction

MRG Effitas is an independent IT security research company, with a heavy focus on applied malware analysis. Besides conventional AV efficacy testing and providing samples to other players in the AV field, we regularly test APT detection appliances and enterprise grade IT security products, simulating realistic attack scenarios. In this regard, testing methods have evolved rapidly over the last couple of years as most labs, under the guidance of AMTSO (of which MRG Effitas is a member) strived to conduct "Real World" testing.

Tests Applied

MRG Effitas performed an in-depth test of several Android AV applications. The level of protection provided was measured in real-life scenarios with in-the-wild pieces of malware as well as some benign samples to map the shortcomings of the applied detection mechanisms. This report summarises the results of our efficacy tests.

Testing took place on Android 6.0.0 Genymotion emulator images in November and December 2018. Though dated, this Android version covers a large portion of user devices in the market. In cases where ARM native libraries have been used and the AV application could not be installed on an x86 emulator, we opted for a stock Nexus 5x device with Android 6.0.0. In order to ensure maximum compatibility for samples that contain native ARM code, the ARM Translation package has also been installed on emulator images.

Our efforts were focused on the following aspects of the products.

Early Stage Detection

Our first scenario focused on an early stage of detection, when test samples have been copied on the SD Card drive of the test device. In the tested scenario, the device has not yet been infected, malicious APK files have only been downloaded, ready to be installed. In our opinion, a properly designed AV suite should detect threats as early as possible and should not allow users to install potentially dangerous applications on their devices.

Detailed steps were as follows.

1. Having prepared the test device, we installed and initialized the AV application (accepted the EULA, downloaded the latest definition files, accepted all requested permissions etc.) When asked, we enabled SD Card scanning features[1]. In cases where we received configuration guides from the vendor, we followed the steps detailed there.
2. We set up the application to include the SD Card in the scan scope.
3. We downloaded the sample set to the SD Card and started the scan.
4. We instructed the application to remove all suspicious files.
5. We ran the scan again, until we saw no warning or suspicious files on the device.
6. We collected the remaining samples.

[1] Due to performance reasons, this option was disabled for most AVs after an out-of-the box initialization.

Detection During Installation

The second scenario involved individual installation of each sample, aiming to check the level of protection provided by the participants.

1. Using adb, we performed an install operation on the device. Following the installation, the AV was informed about the newly installed application, kicking in detection routines.
2. We gave plenty of time for the AV to finish all scanning activities [2,3].
3. We created a screen shot of the resulting screen. Should the AV display a warning or an alert, the test was counted as a Pass, no warning resulted in a Miss. All logcat logs were saved from the device during the process.
4. Using adb, we uninstalled the sample and went on to test the next one.

Note that on Android, installation of a piece of malware does not necessarily mean unwanted consequences for the user, as it is the first launch that kicks in actual malicious code. Having started the sample however, can have detrimental consequences from a security perspective. After the first launch, a piece of malware requesting SYSTEM_ALERT_WINDOW permission is able to continuously display a Device Administrator or an Accessibility Admin request screen to the user. In such cases, the user is unable to get rid of the application as they have no access to the launcher, the application drawer or the Settings application to perform an uninstall[4].

False Positive Tests

In order to cover all aspects of the efficacy of the participants, a limited set of samples has also been selected. The samples have been downloaded from a well-known 3rd party app store, exhibiting no malicious behaviour but requiring a varying range of permissions.

The samples have been selected to cover the following categories.

- Benign samples re-signed using a freshly generated, ‘neutral’ certificate.
- Benign samples signed with their original developer certificate.

Samples

Malicious In-the-wild Samples

Testing used an initial 228-sample malware set. All samples have been categorized using the following labels.

- **SMS Payment.** The application provides features to send SMS messages to premium rate numbers. Most of the selected samples were able to ‘auto-send’ messages, as they opted for the SEND_SMS permission, resulting in a direct financial loss for the victim.

[2] The timeout threshold is a critical aspect of testing. Should the value too low, the test results do not reflect actual results as the AV has no chance of finishing detection. We aim to choose the threshold to be realistic, as it is unlikely that a user waits for several minutes after installation before actually starting the newly installed application – in our testing methodology, a ‘too late’ detection or a detection without a clear notification is also considered a Miss.

[3] During the result discussion stage, we actively cooperate with vendors to eliminate timeout related issues, in order to make sure that the figures presented in the report reflect the results of a realistic scenario.

[4] Note that in order to mitigate this kind of typical malware behaviour, the Android API design team reviewed the Device Administrator and the Accessibility Admin Request screens to include a checkbox that can be used to prevent the OS from displaying the screen again. This feature however, made its way only to recent revisions of the Android API.

- **Trojan.** Trojans are applications, which display a certain set of features within their description. However, the implemented modules require a wide range of permissions, which do not belong to the advertised functionality. A typical example is a flashlight app, which can read the contact list, the GPS position and send them to the Internet.
- **Spyware.** We classified a sample Spyware if it leaks information, which can be used to track the user (as most security-conscious users do not wish to be tracked). Ironically, most ad propelled applications using aggressive frameworks qualify as spyware, as they leak IMEI, phone number, phone vendor and model etc. to the ad provider network.
- **Financial/banking.** This type of malware aims for direct financial abuse. A typical financial piece of malware detects if the user is logged in to a mobile banking session using either a browser or mobile banking application and, for instance, might attempt to display a matching phishing site or to draw an overlay window to fool the user into thinking that the session has ended and that they need to re-authenticate. Typically, such samples use permissions to get the task list, combined with the SYSTEM_ALERT_WINDOW permission.
- **PUA.**[5] The term ‘Potentially Unwanted Applications’ denotes applications, which perform actions that are not in alignment with the security-conscious user’s intentions. For instance, applications provided with aggressive advertisement modules usually make it possible for ad campaigners to track individual users, even to assign the device with the user’s demographic properties through social network ad services. Effitas claims that security-conscious users are sensitive regarding their privacy and possibly no application feature can make it up for the users’ private data and browsing habits to be sold over the Internet and a decent AV should let the user know if such an application is about to be installed.

Note that most samples implement several kinds of operation, therefore most samples fall into several categories (for instance, consider a typical piece of malware, which serves malicious ads and if possible, it attempts to obtain the SEND_SMS permission to send premium rate messages).

Figure 1 depicts the distribution of test samples.

[5] Android applications with a social network integrated advertising module often fall into a kind of ‘grey zone’ from a detection perspective, as any application can be turned into a PUA, should the developers include an aggressive advertising module. Hence, we included charts, which handle PUA and non-PUA samples separately.

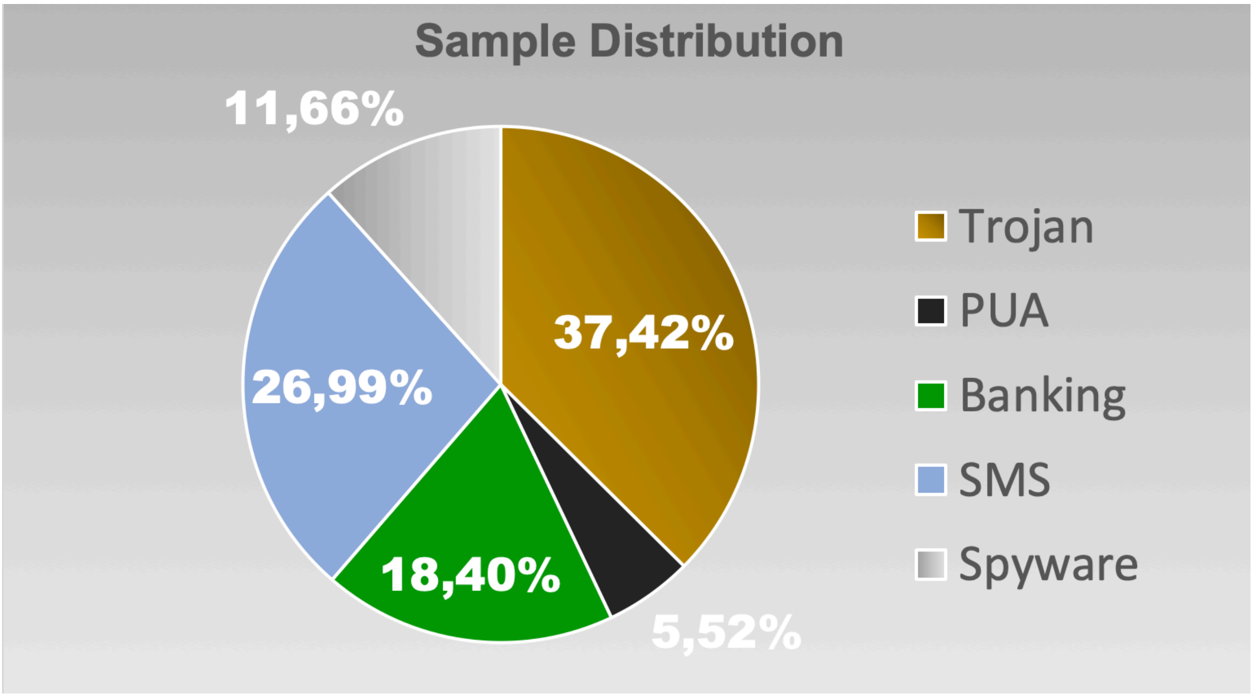


Figure 1 - In-the-wild malware set distribution

Simulator samples

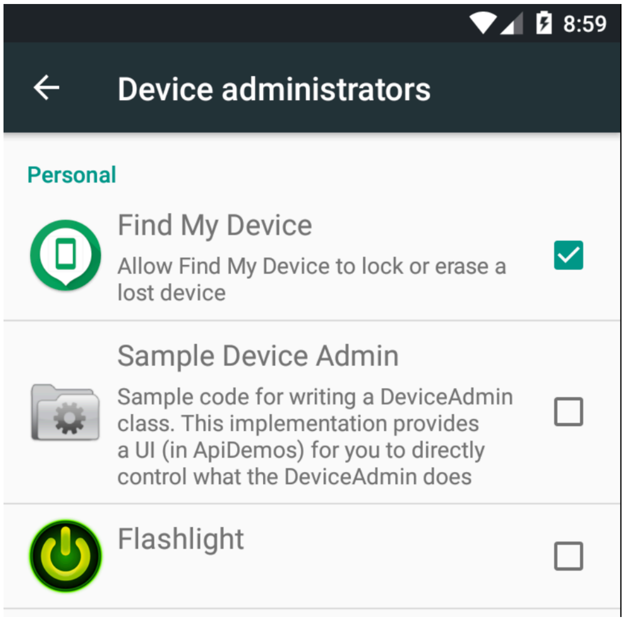
Simulators are custom samples, introduced into the testing process to put the sophistication of the detection routines to the test. Our simulators were created to simulate the attack model of a 'malicious 3rd party app store providing backdoored applications' type of scenario, which means that counterfeit versions of legitimate applications are provided to the victims (many times pirated application versions can be downloaded for free-of-charge). The counterfeit versions are backdoored versions of popular applications, which, while retaining the functionality of the original application, also include malicious modules.

The samples have been created using a proof-of-concept engine using static smali byte code injection techniques, making no effort to obscure the malicious actions of the injected modules. Many of the simulator samples have been modified to implement Device Administrator features, which is a common trait for several malware families.

For testing, we used 9 custom created samples. Our custom samples performed one or more of the following 'extra' operations during each start-up of the main activity handling the LAUNCHER intent.

1. Monitoring and sending the SMS list to a custom Internet endpoint
2. Leaking IMEI, IMSI, phone model to a custom Internet endpoint
3. Opting for the Device Administrator privilege[6]
4. Automated sending of SMS messages

[6] We selected applications, which normally do not utilise this feature



8:59

Device administrators

Personal

Find My Device ☑

Allow Find My Device to lock or erase a lost device

Sample Device Admin □

Sample code for writing a DeviceAdmin class. This implementation provides a UI (in ApiDemos) for you to directly control what the DeviceAdmin does

Flashlight □

Figure 2 - A counterfeit device administrator application

False positive samples

For false positive testing, an 18-sample set was used. Their distribution with regards to developer certificate was as follows.

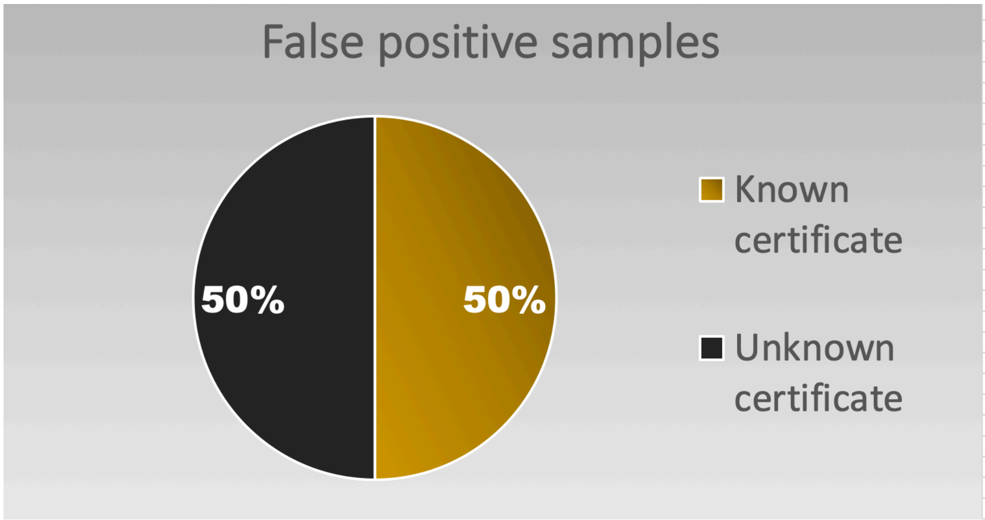


Figure 3 - Distribution of signatures of non-malicious samples

Security Applications Tested

The following security suites have been selected for testing.

Vendor	Name	Play Store identifier	Version
Antivirus Pro	Antivirus Pro for Android	com.protoolapps.antivirus.security.android	1.4.1
AV Cleaner	Antivirus Cleaner	antivirus.cleaner.phoneboost[7]	1.1
Avast	Avast Mobile Security & Antivirus	com.avast.android.mobilesecurity	6.15.1
AVG	Antivirus free	com.antivirus	6.15.1
ESET	Mobile Security and Antivirus	com.eset.ems2.gp	4.3.7.0
Kaspersky	Kaspersky Mobile Antivirus	com.kms.free	11.18.4.905
McAfee	McAfee Mobile Security	com.wsandroid.suite	5.0.2.1839
NDAntivirus	Antivirus Free	ndtools.antivirusfree	1.3
Powersecurity	Power Security - Anti Virus & Phone Cleaner	com.lm.powersecurity	2.1.2
Symantec	Norton Security and Antivirus	com.symantec.mobilesecurity	4.4.0.4302
Zoner	Zoner Antivirus	com.zoner.android.antivirus	1.14.1

Table 1 – The list of selected participants

Test Results

The tables and charts below show the results of testing under the MRG Effitas Android AV Testing Program.

[7] Note that the application has been removed from the Play Store during testing.

In-The-Wild Tests

Overall non-PUA Detection

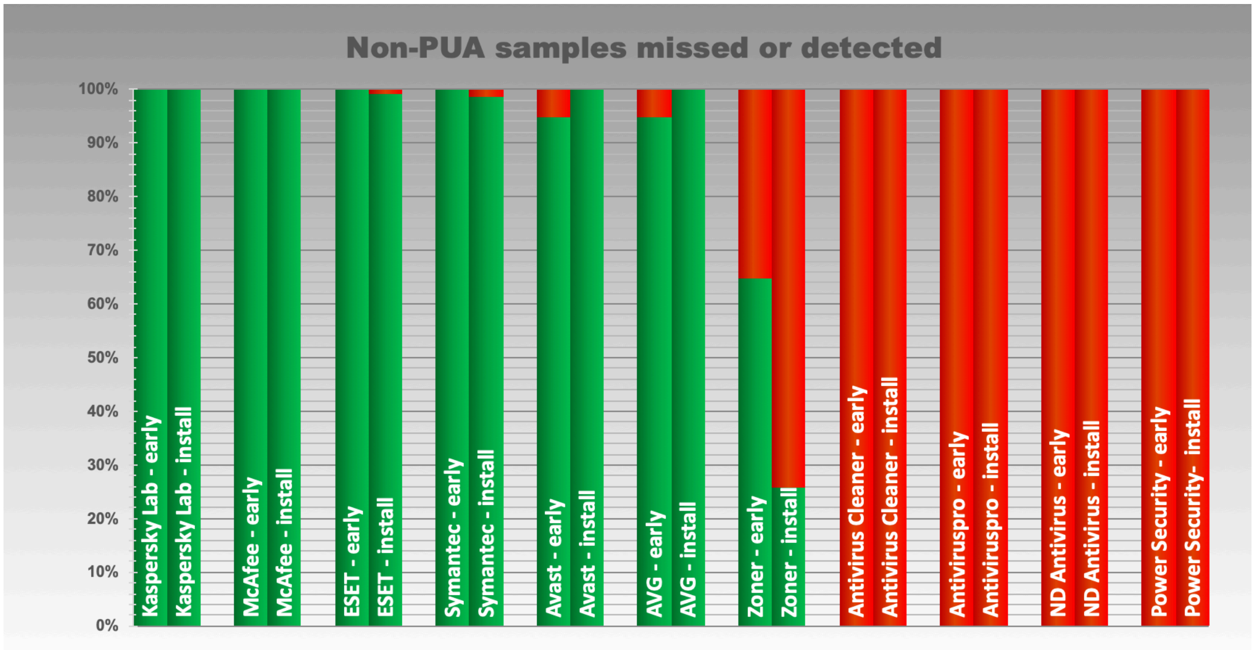


Figure 4 - Summary, non-PUA samples

Category: Summary, Non-PUA Samples								
	Early				Install			
Participant	Early blocked		Early missed		Install blocked		Install missed	
Kaspersky Lab	210	100%	0	0%	210	100%	0	0%
McAfee	210	100%	0	0%	210	100%	0	0%
ESET	210	100%	0	0%	208	99,05%	2	0,95%
Symantec	210	100%	0	0%	207	98,57%	3	1,43%
Avast	199	94,76%	11	7,89%	210	100%	0	0%
AVG	199	94,76%	11	7,89%	210	100%	0	0%
Zoner	136	64,76%	74	28,95%	54	25,71%	156	74,29%
Antivirus Cleaner	0	0%	210	100%	0	0%	210	100%
Antivirus Pro	0	0%	210	100%	0	0%	210	100%
ND Antivirus	0	0%	210	100%	0	0%	210	100%
Power Security	0	0%	210	100%	0	0%	210	100%

Table 2 – Results, non-PUA samples

PUA detection

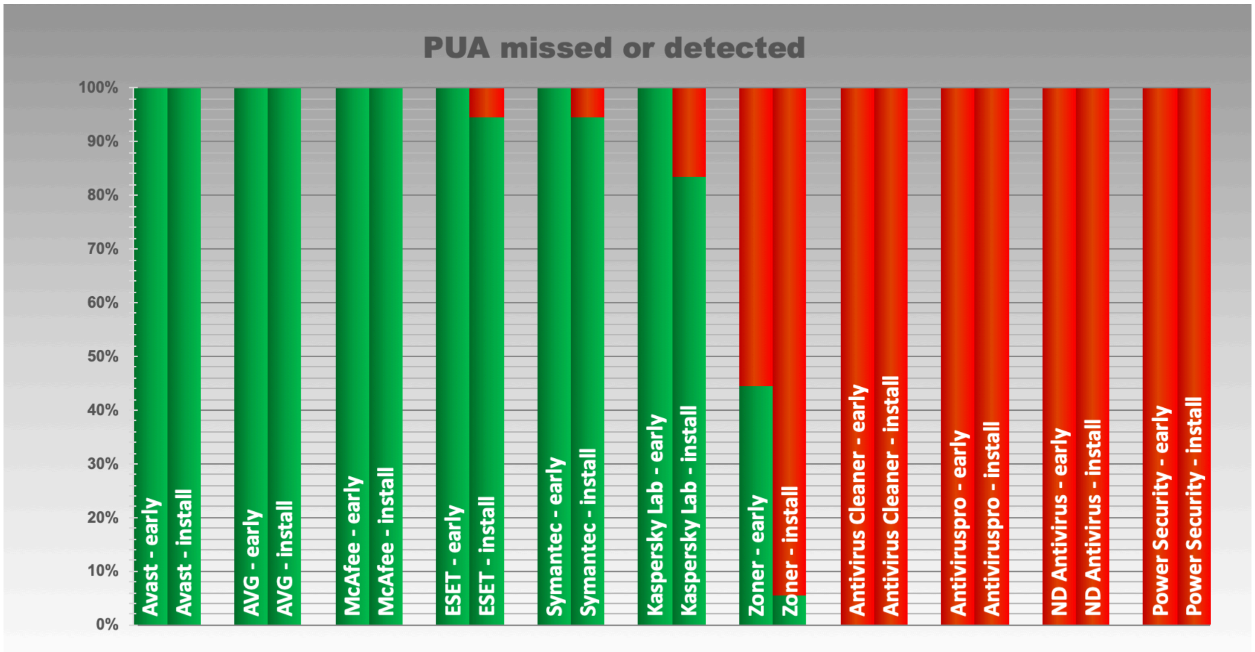


Figure 5 - Summary, PUA samples

Category: PUA Samples								
	Early				**Install**			
Participant	**Early blocked**		**Early missed**		**Install blocked**		**Install missed**	
Avast	18	100%	0	0%	18	100%	0	0%
AVG	18	100%	0	0%	18	100%	0	0%
McAfee	18	100%	0	0%	18	100%	0	0%
ESET	18	100%	0	0%	17	94,44%	1	5,56%
Symantec	18	100%	0	0%	17	94,44%	1	5,56%
Kaspersky Lab	18	100%	0	0%	15	83,33%	3	16,67%
Zoner	8	44,44%	10	55,56%	1	5,56%	17	94,44%
Antivirus Cleaner	0	0%	18	100%	0	0%	18	100%
Antivirus Pro	0	0%	18	100%	0	0%	18	100%
ND Antivirus	0	0%	18	100%	0	0%	18	100%
Power Security	0	0%	18	100%	0	0%	18	100%

Table 3 – Results, PUA samples

Trojan detection

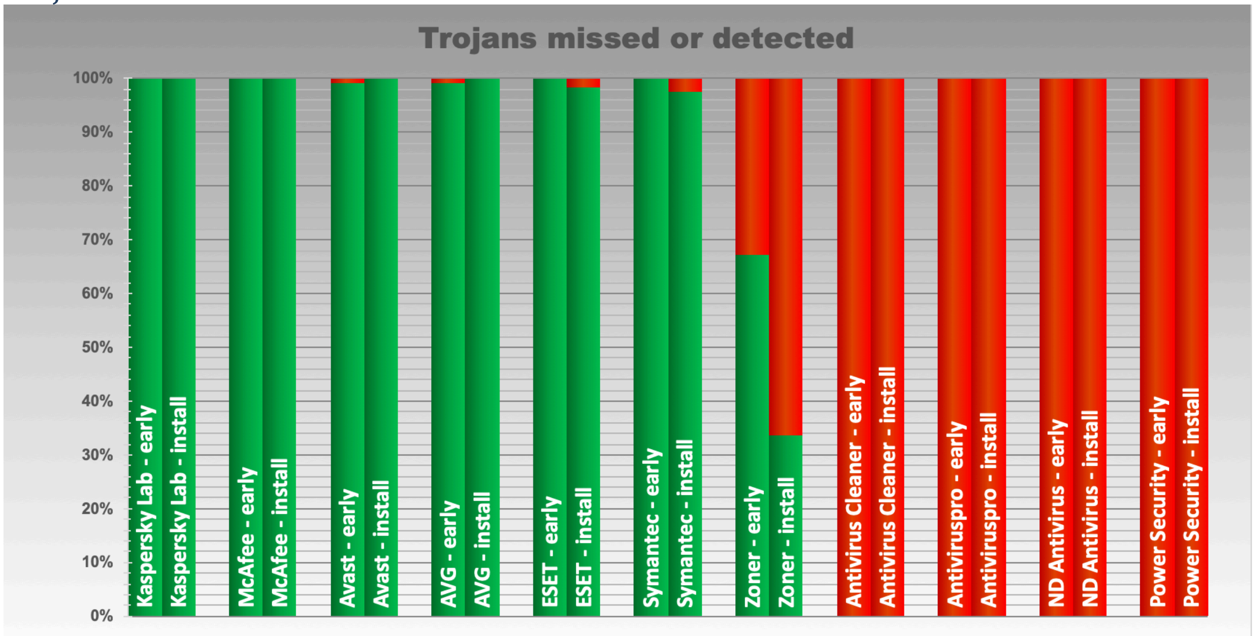


Figure 6 - Summary, trojan samples

Category: Trojan Samples								
	Early				Install			
Participant name	Detected		Missed		Detected		Missed	
Kaspersky Lab	122	100%	0	0%	122	100%	0	0%
McAfee	122	100%	0	0%	122	100%	0	0%
Avast	121	99,18%	1	0,82%	122	100%	0	0%
AVG	121	99,18%	1	0,82%	122	100%	0	0%
ESET	122	100%	0	0%	120	98,36%	2	1,64%
Symantec	122	100%	0	0%	119	97,54%	3	2,46%
Zoner	82	67,21%	40	32,79%	41	33,61%	81	66,39%
Antivirus Cleaner	0	0%	122	100%	0	0%	122	100%
Antivirus Pro	0	0%	122	100%	0	0%	122	100%
ND Antivirus	0	0%	122	100%	0	0%	122	100%
Power Security	0	0%	122	100%	0	0%	122	100%

Table 4 – Results, trojan samples

SMS detection

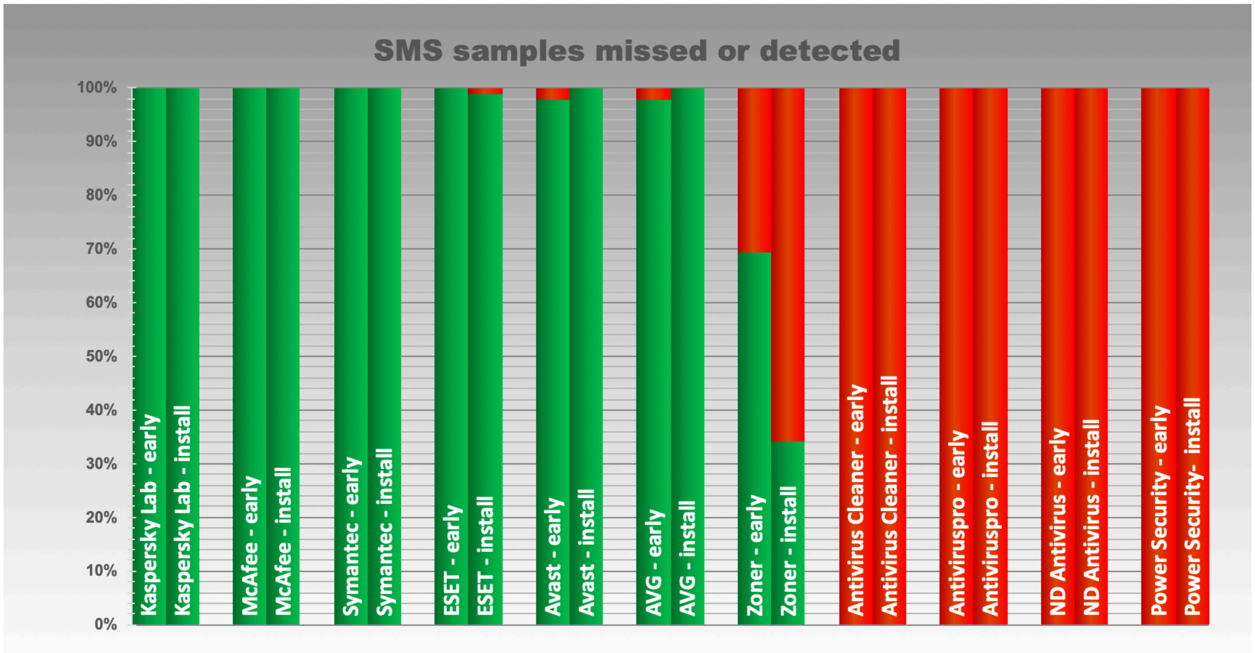


Figure 7 - Summary, SMS samples

Category: SMS Samples								
	Early				Install			
Participant name	Detected		Missed		Detected		Missed	
Kaspersky Lab	88	100%	0	0%	88	100%	0	0%
McAfee	88	100%	0	0%	88	100%	0	0%
Symantec	88	100%	0	0%	88	100%	0	0%
ESET	88	100%	0	0%	87	98,86%	1	1,14%
Avast	86	97,73%	2	2,27%	88	100%	0	0%
AVG	86	97,73%	2	2,27%	88	100%	0	0%
Zoner	61	69,32%	27	30,68%	30	34,09%	58	65,91%
Antivirus Cleaner	0	0%	88	100%	0	0%	88	100%
Antivirus Pro	0	0%	88	100%	0	0%	88	100%
ND Antivirus	0	0%	88	100%	0	0%	88	100%
Power Security	0	0%	88	100%	0	0%	88	100%

Table 5 – Results, SMS samples

Spyware detection

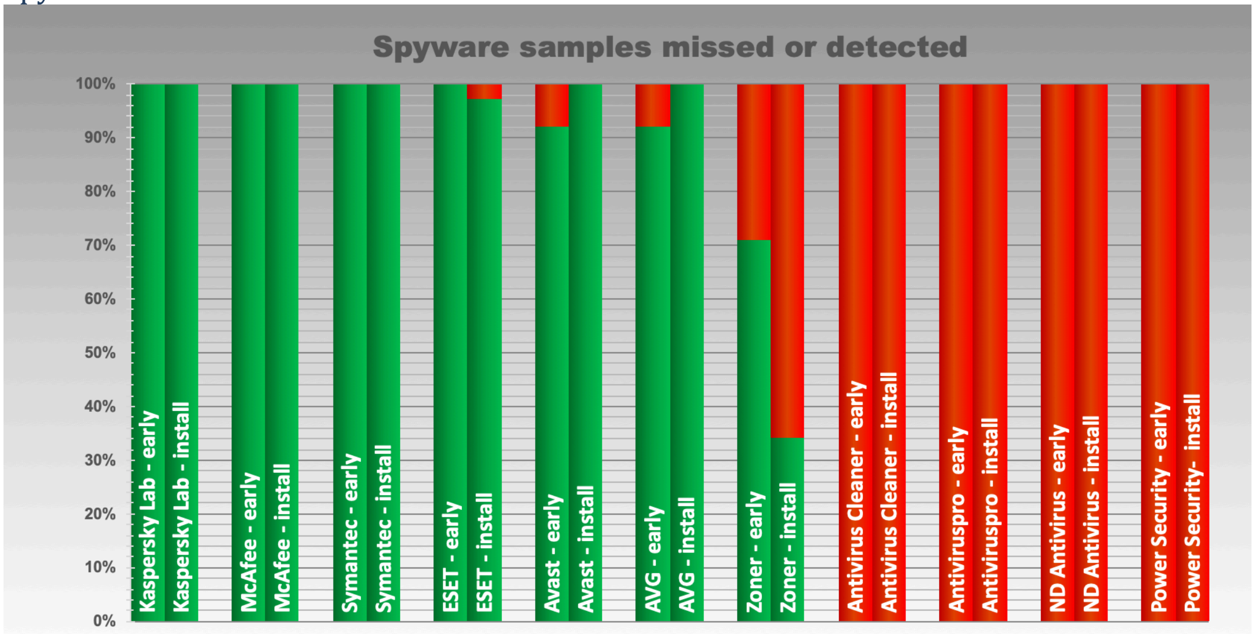


Figure 8 - Summary, Spyware samples

Category: Spyware samples								
	Early				Install			
Participant name	Detected		Missed		Detected		Missed	
Kaspersky Lab	38	100%	0	0%	38	100%	0	0%
McAfee	38	100%	0	0%	38	100%	0	0%
Symantec	38	100%	0	0%	38	100%	0	0%
ESET	38	100%	0	0%	37	97,37%	1	2,63%
Avast	35	92,11%	3	7,89%	38	100%	0	0%
AVG	35	92,11%	3	7,89%	38	100%	0	0%
Zoner	27	71,05%	11	28,95%	13	34,21%	25	65,79%
Antivirus Cleaner	0	0%	38	100%	0	0%	38	100%
Antivirus Pro	0	0%	38	100%	0	0%	38	100%
ND Antivirus	0	0%	38	100%	0	0%	38	100%
Power Security	0	0%	38	100%	0	0%	38	100%

Table 6 – Results, spyware samples

Banking detection

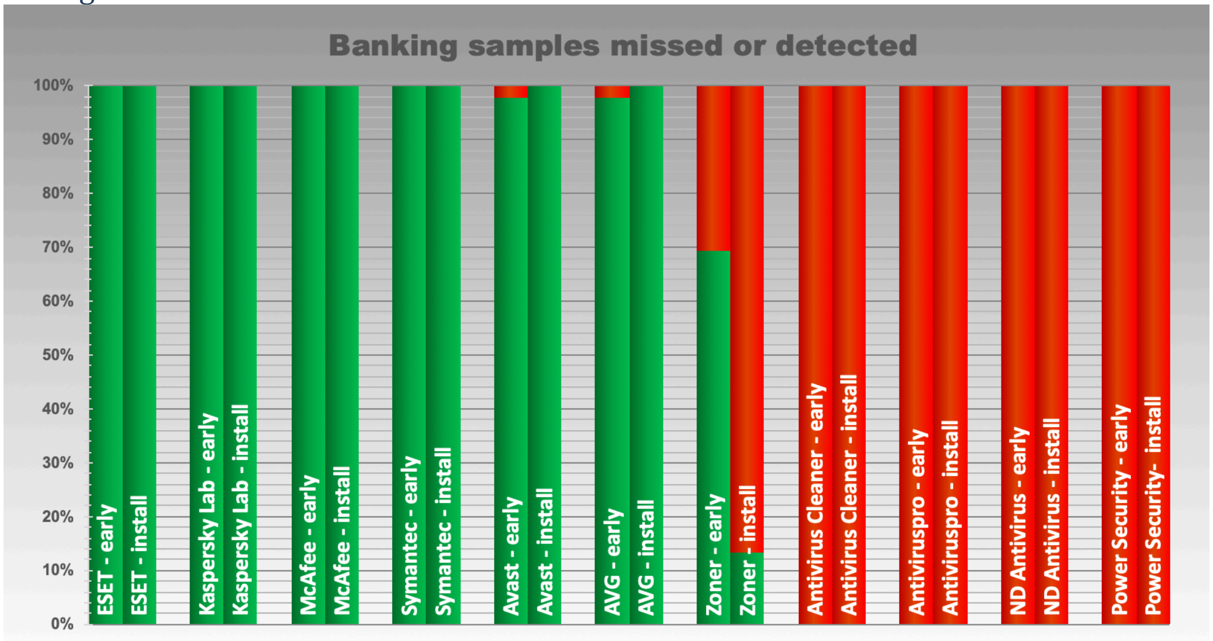


Figure 9 - Summary, banking samples

Category: Banking samples								
	Early				Install			
Participant name	Detected		Missed		Detected		Missed	
ESET	60	100%	0	0%	60	100%	0	0%
Kaspersky Lab	60	100%	0	0%	60	100%	0	0%
McAfee	60	100%	0	0%	60	100%	0	0%
Symantec	60	100%	0	0%	60	100%	0	0%
Avast	52	97,73%	8	2,27%	60	100%	0	0%
AVG	52	97,73%	8	2,27%	60	100%	0	0%
Zoner	33	69,32%	27	30,68%	8	13,33%	52	86,67%
Antivirus Cleaner	0	0%	60	100%	0	0%	60	100%
Antivirus Pro	0	0%	60	100%	0	0%	60	100%
ND Antivirus	0	0%	60	100%	0	0%	60	100%
Power Security	0	0%	60	100%	0	0%	60	100%

Table 7 – Results, banking samples

Simulator tests

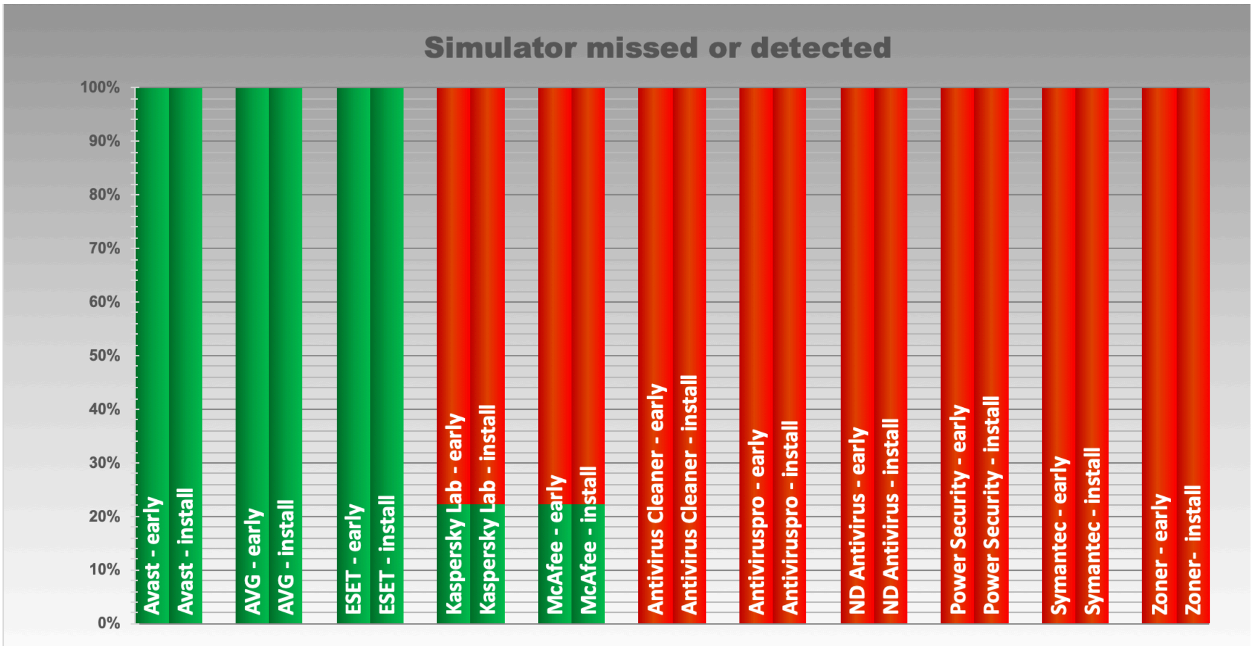


Figure 10 - Summary, simulator samples

Category: Simulator samples								
	Early				Install			
Participant name	Detected		Missed		Detected		Missed	
Avast	9	100%	0	0%	9	100%	0	0%
AVG	9	100%	0	0%	9	100%	0	0%
ESET	9	100%	0	0%	9	100%	0	0%
Kaspersky Lab	2	22,22%	7	77,78%	2	22,22%	7	77,78%
McAfee	2	22,22%	7	77,78%	2	22,22%	7	77,78%
Antivirus Cleaner	0	0%	9	100%	0	0%	9	100%
Antiviruspro	0	0%	9	100%	0	0%	9	100%
ND Antivirus	0	0%	9	100%	0	0%	9	100%
Power Security	0	0%	9	100%	0	0%	9	100%
Symantec	0	0%	9	100%	0	0%	9	100%
Zoner	0	0%	9	100%	0	0%	9	100%

Table 8 – Results, simulator samples

False Positive test

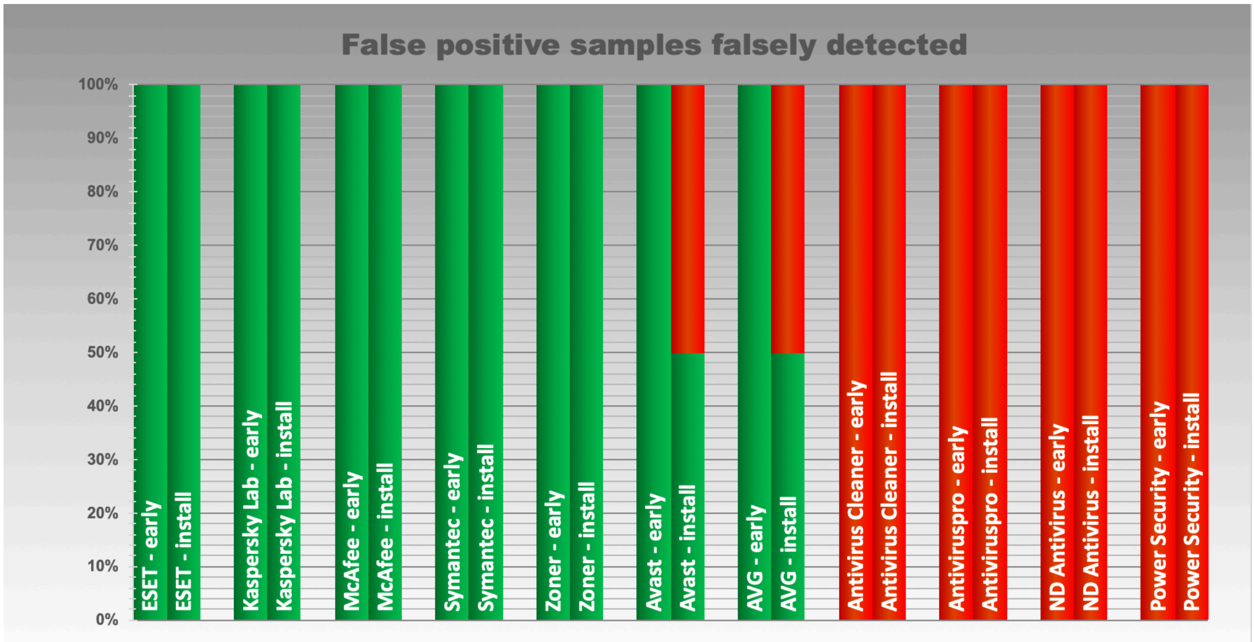


Figure 11 - Summary, benign samples

Category: Benign samples								
	Early				Install			
Participant name	Passed (no alert)		Missed		Passed (no alert)		Missed	
ESET	18	100%	0	0%	18	100%	0	0%
Kaspersky Lab	18	100%	0	0%	18	100%	0	0%
McAfee	18	100%	0	0%	18	100%	0	0%
Symantec	18	100%	0	0%	18	100%	0	0%
Zoner	18	100%	0	0%	18	100%	0	0%
Avast	18	100%	0	0%	9	50%	9	50%
AVG	18	100%	0	0%	9	50%	9	50%
Antivirus Cleaner	0	0%	18	100%	0	0%	18	100%
Antiviruspro	0	0%	18	100%	0	0%	18	100%
ND Antivirus	0	0%	18	100%	0	0%	18	100%
Power Security	0	0%	18	100%	0	0%	18	100%

Table 9 – Results, benign samples

Summary

Results

As a result of testing, the following AV engines scored a 100% detection rate in the early testing scenario of non-PUA samples.

- Kaspersky Lab
- McAfee
- ESET
- Symantec

The following AV engines reached detection rates between 90% and 100% in the early detection of non-PUA samples scenario.

- Avast
- AVG

The following AV engines scored 100% detection rate in an after-installation detection scenario of non-PUA samples.

- Kaspersky Lab
- McAfee
- Avast
- AVG

The following AV engines reached detection rates between 90% and 100% in an after-installation detection of non-PUA samples scenario.

- ESET
- Symantec

Conclusions

As a result of our testing efforts, a couple of conclusions can be drawn from our time with the AV engines and samples in our test lab.

‘AV as another app’

Testing led us to the conclusions that detection for most AVs relies heavily on the metadata of installed packages (hashes, developer certificates etc.), meaning that unlike in a Windows based environment, an AV is unable to get an insight into the actual activity of other applications. This behaviour is in alignment with the basic Android security principles, handling an AV as ‘just another app’[8].

Detection mechanisms

Our tests confirmed that most AVs use different methods for detection before and after installation. This is due to the fact that prior to installation, different set of metadata is available for and AV engine of a file that is stored on the SD card than what is available after its installation.

[8] For further insight on the topic, see our blog post on https://www.mrg-effitas.com/research/android-av-vs-third-party-app-stores/

Simulator detection

Most AV engines, having detected our custom simulator samples in past tests, were able to perform detection purely based on the package signature traits. This means that even though a notification has been displayed for those samples, the successful detection has been a result of a mechanism, heavily prone to false positives. As a result, in our previous Android 360 engagements many AVs had problems with detecting the simulator samples.

We were glad to see significant improvements in this regard, namely ESET was able to pinpoint all our crafted samples, scoring a perfect 100% in the simulator test with no side effect issues in False Positive tests.

Avast and AVG also detected our simulator samples as suspicious applications, however False Positive tests proved that the detection was not based on any behaviour traits but on signature analysis, which, while being a valid approach with its own merits, is also prone to false positives.

Counterfeit AV applications[9]

For the first time in Android 360 history, a couple of new participants have been selected. These apps have been selected using a simple Play Store search for 'Android AV'. As it turned out, many the selected applications performed significantly worse than any of the products from well-known vendors. As the below four applications did not even thrive to aid the user for improving their device security, we scored all engines at 0% in all detection categories. The applications have been reported as inappropriate to Google.

- **Antivirus Cleaner**. Did not perform any kind of AV detection activity, upon install, a warning screen has been displayed to the user, regardless to the nature of the sample.
- **Antivirus Pro**. Did not perform any kind of AV detection activity.
- **ND Antivirus.** Should the application be installed from a non-Play Store source, a warning screen is displayed listing the requested permissions, regardless to the nature of the sample.
- **Power Security**. Did not perform any actual AV activity. Furthermore, upon install, an 'application is safe' screen has been displayed to the user, containing advertisements.

[9] For details, refer to page 7.