Android 360° Assessment Programme

Q2 2021

MRG Effitas
EFFICACY ASSESSMENT & ASSURANCE

MRG Effitas Ltd.

Effitas is a world-leading, independent IT security efficacy testing & assurance company. We are trusted by antimalware vendors across the world.

MANAGEMENT TEAM:

- Chris Pickard, Chief Executive Officer
- Norbert Biro, Chief Technical Officer

WEBSITE:
www.mrg-effitas.com

TEL:
+44 (0)20 3239 9289

EMAIL:
contact@mrg-effitas.com

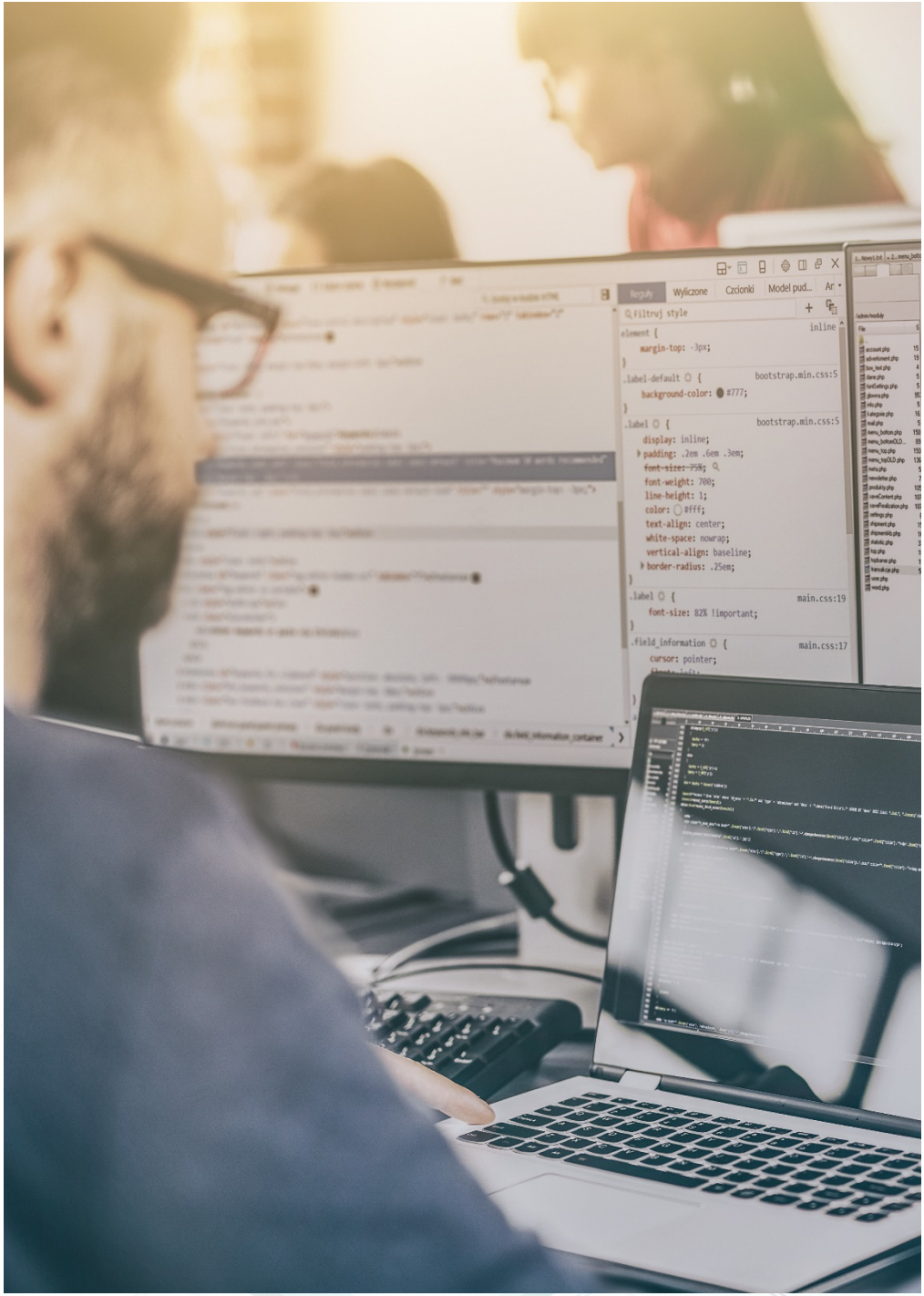
TWITTER:
@mrgeffitas

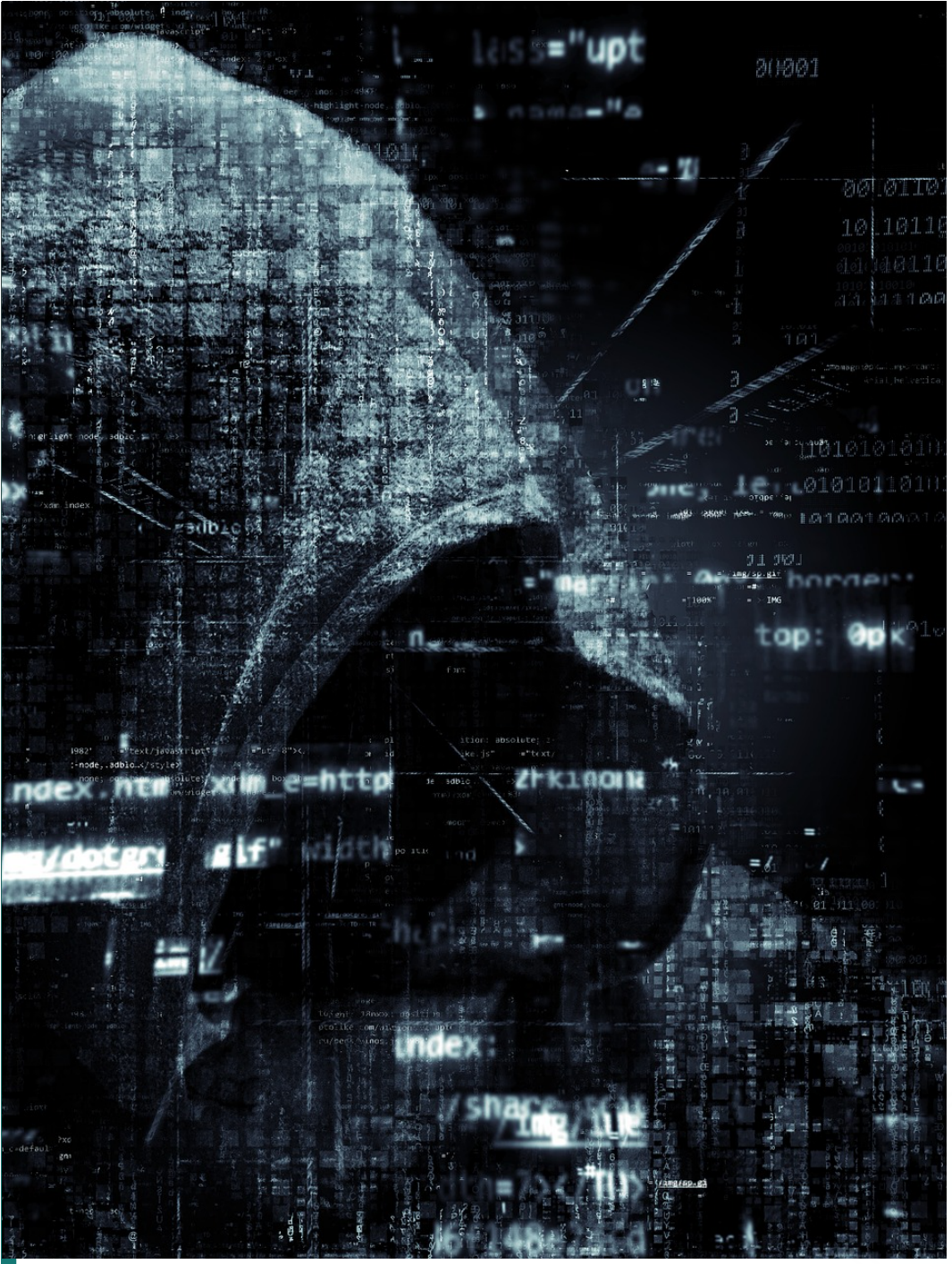
Contents

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.

Android devices are used by over 2.3 Billion people around the globe. As the overall platform philosophy allows an easy-to-opt in platform with no mandated central application distribution platform, Android based malware has been on a constant rise since the early Gingerbread days. As a result, the market for Android AVs is heaving with applications that promise loud taglines with '100% security'. A quick search on the Play Store for Antivirus products reveals literally hundreds of results – our test aims to help user decisions with a complex test regime with both in-the-wild and artificially crafted simulator samples and results that reflect a real-life efficacy of our test participants.





Our Mission

In providing quarterly certifications, the MRG Effitas 360 Programme is the de facto standard by which security vendors, financial institutions and other corporations can attain the most rigorous and accurate determination of a product's efficacy against current financial malware attacks.

We test over twelve months beginning in Quarter 2 and ending in Quarter 1, at which point (or shortly after) we publish our results. As with all our certification testing, we work with vendors, offering feedback and helping them to improve their product as we go.

Products that pass all tests during a quarter will receive the MRG Effitas certification for Android efficacy protection.

More information about the compliance status of this test can be found on the AMTSO website.

https://www.amtso.org/tests/mrg-effitas-q2-2021-360-android-assessment-and-certification/

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 8.0.0 Genymotion emulator images in May and February 2021, covering a significant portion of user devices on the market. In order to ensure maximum compatibility for samples that contain native ARM code, the ARM Translation package has also been installed on emulator images. In cases where ARM native libraries have been extensively used and the AV application could not be installed or properly run on an x86 emulator, we opted for stock Nexus 5x devices with Android 8.0.0. In order to ensure the cleanliness of testing process, the Play Protect feature has been disabled.

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.

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 screenshot of the resulting screen. Should the AV display a warning or an alert, the test was counted as a Pass, no

[1] Due to performance reasons, this option was disabled for most AVs after an out-of-the box initialization.
[2] The timeout threshold is a critical aspect of testing. Should the value be too low, the test results would 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.

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 any actual malicious code within. 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.

Samples

Malicious In-the-wild Samples

Testing used an initial 160-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 usually opted for the SEND_SMS permission, resulting in a direct financial loss for the victim.
- **Trojan.** Trojans are applications, which display a certain set of features within their description and their overall appearance suggests some expectations regarding their functionality. However, the implemented modules require a wider range of permissions, which do not belong to the advertised functionality. A typical example is a flashlight app, which can read the contact list, location information 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-

[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.

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. 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).

Our ITW samples are obtained using several sources. As we thrive to use fresh pieces, the actual mix, utilised in the test batches, is highly dependent on the then-current activity of malicious threat actors. As a result, in many batches, one or more families were represented with several samples (all of them being fresh at the time). Consequently, failing to detect that family in the affected batches, will result in multiple Misses.

Figure 1. depicts the distribution of test samples.

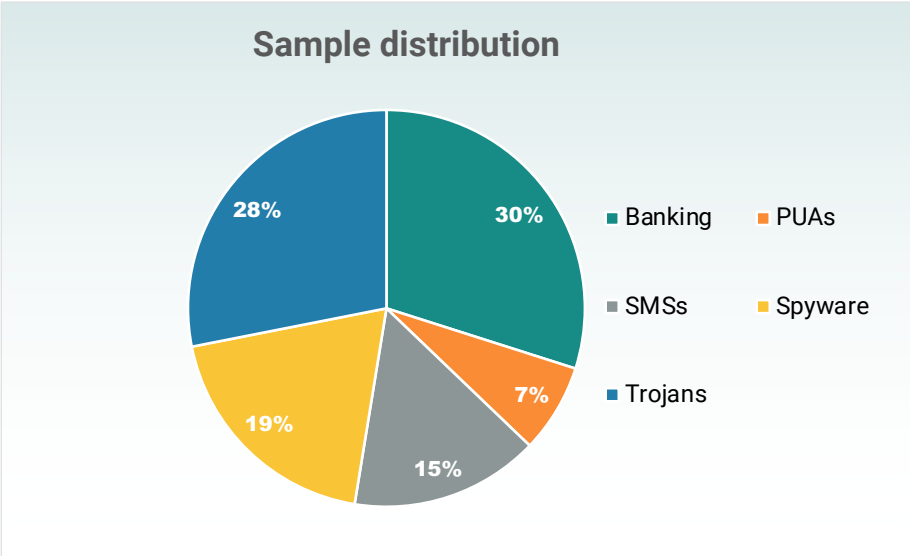


FIGURE 1. IN-THE-WILD SAMPLE DISTRIBUTION

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

[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.

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 Accessibility features, which is a common trait for several malware families.

For testing, we used 5 custom created samples. It is important to stress that these samples have not been collected or observed in-the-wild. Our custom samples implemented a well-known method exploiting the accessibility features of the Android API, which has been a popular method to read on-screen messages, SMS tokens, banking details and other sensitive information. Our samples were counterfeit versions of legitimate Android applications, sending SMS messages, keystrokes, passwords etc. to our custom HTTP web service endpoint.

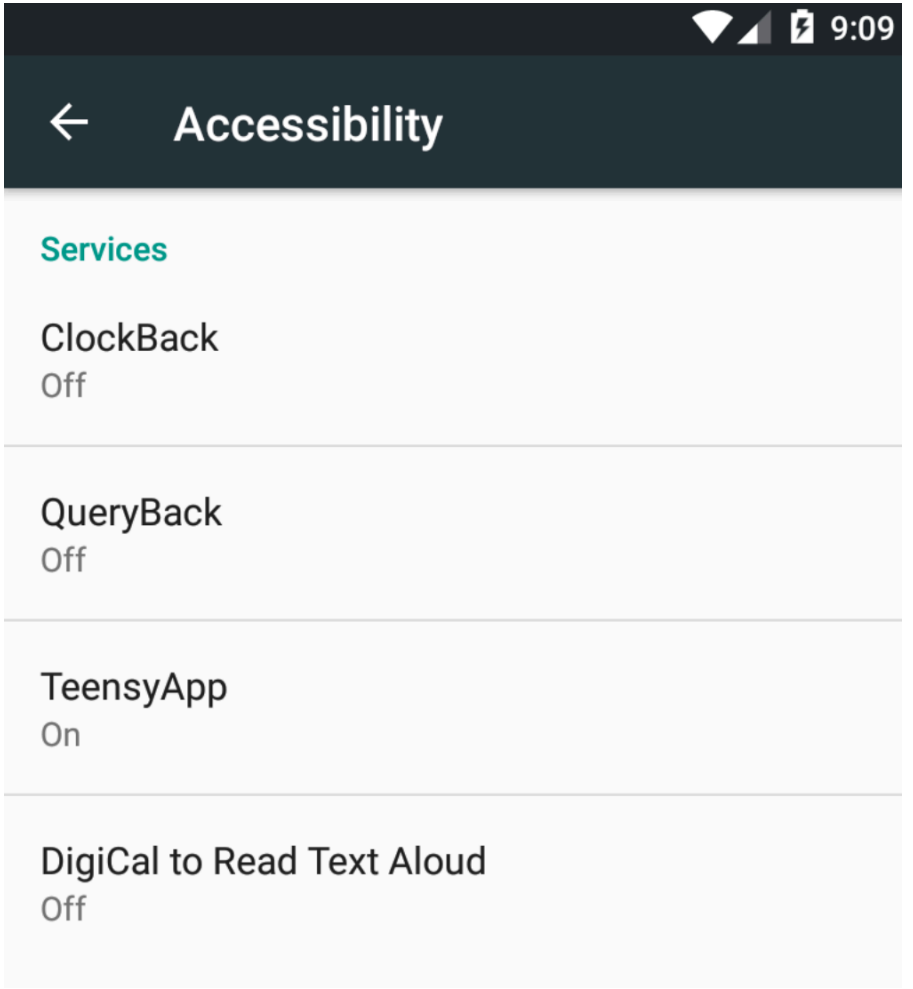


FIGURE 2. COUNTERFEIT APPLICATIONS EXPLOITING THE ACCESSIBILITY FEATURES

False positive samples

For false positive testing, a 10-sample set was used, retrieved from non-malicious 3rd party applications stores. The applications have been selected to cover a wide range of permissions and functionality.

Security Applications Tested

The following security suites have been selected for testing. Besides well-established vendors with considerable reputation and track history, we select smaller vendors with less market share. As the Play Store is heaving with Android security applications, we tend to select AV products with a considerable number of downloads.

Product name	Caption	Play Store URL
AVG	AVG	https://play.google.com/store/apps/details?id=com.antivirus
Avira Mobile Antivirus	Avira	https://play.google.com/store/apps/details?id=com.avira.android
Bitdefender Mobile Security & Antivirus	Bitdefender	https://play.google.com/store/apps/details?id=com.bitdefender.security
Comodo Mobile Security Antivirus	Comodo	https://play.google.com/store/apps/details?id=com.comodo.cisme.antivirus
Mobile Security & Antivirus	ESET	https://play.google.com/store/apps/details?id=com.eset.ems2.gp
Virus Cleaner, Anti-Malware	Malwarebytes	https://play.google.com/store/apps/details?id=org.malwarebytes.antimalware
Norton Security and Antivirus	Norton	https://play.google.com/store/apps/details?id=com.symantec.mobilesecurity
Zemana Antivirus: Anti-Malware & Web Security	Zemana	https://play.google.com/store/apps/details?id=com.zemana.msecurity
Zoner Antivirus	Zoner	https://play.google.com/store/apps/details?id=com.zoner.android.antivirus

TABLE 1. TEST PARTICIPANTS

Test Results

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

Averaged non-PUA Detection Scores[6]

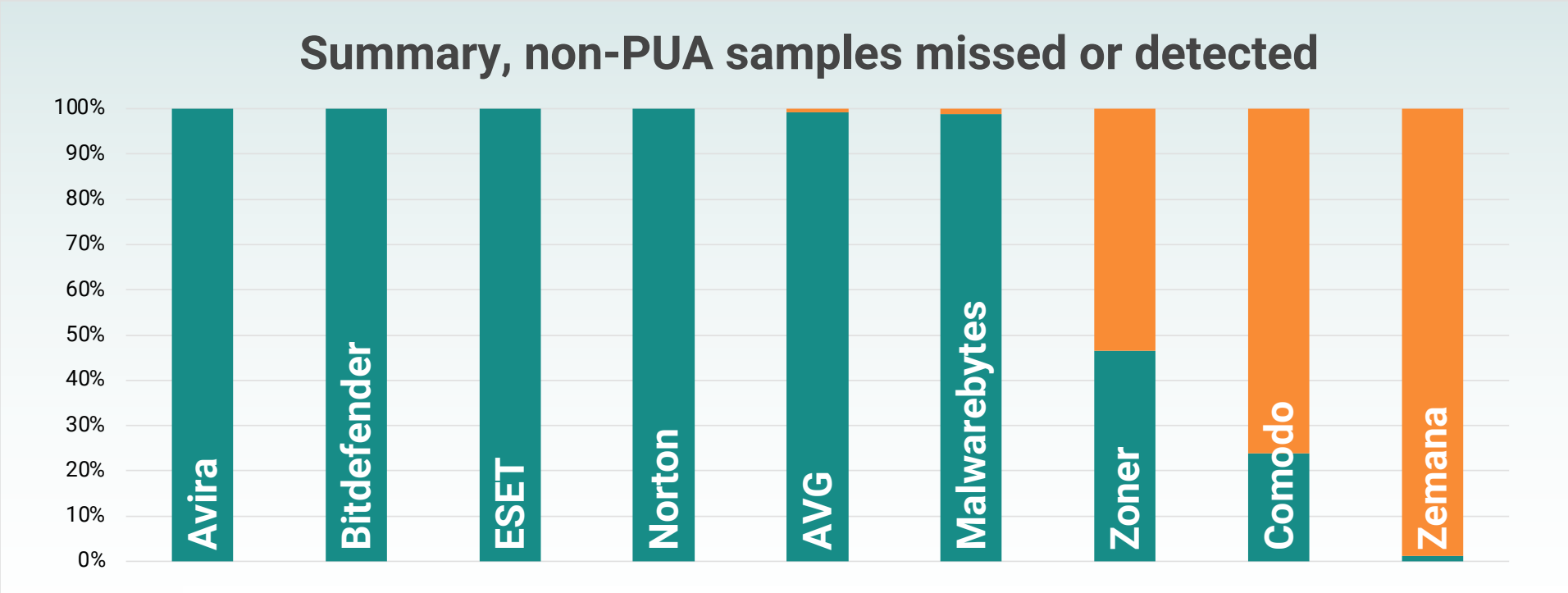


FIGURE 3. SUMMARY, NON-PUA SAMPLES

[6] The figures were created by averaging Early and Install scores for all non-PUA samples (where applicable).

Summary, averaged scores		
Short name	Summary blocked	Summary missed
Avira	100%	0%
Bitdefender	100%	0%
ESET	100%	0%
Norton	100%	0%
AVG	99%	1%
Malwarebytes	99%	1%
Zoner	47%	53%
Comodo	24%	76%
Zemana	1%	99%

TABLE 2. SUMMARISED RESULTS, NON-PUA SAMPLES

Overall non-PUA Detection

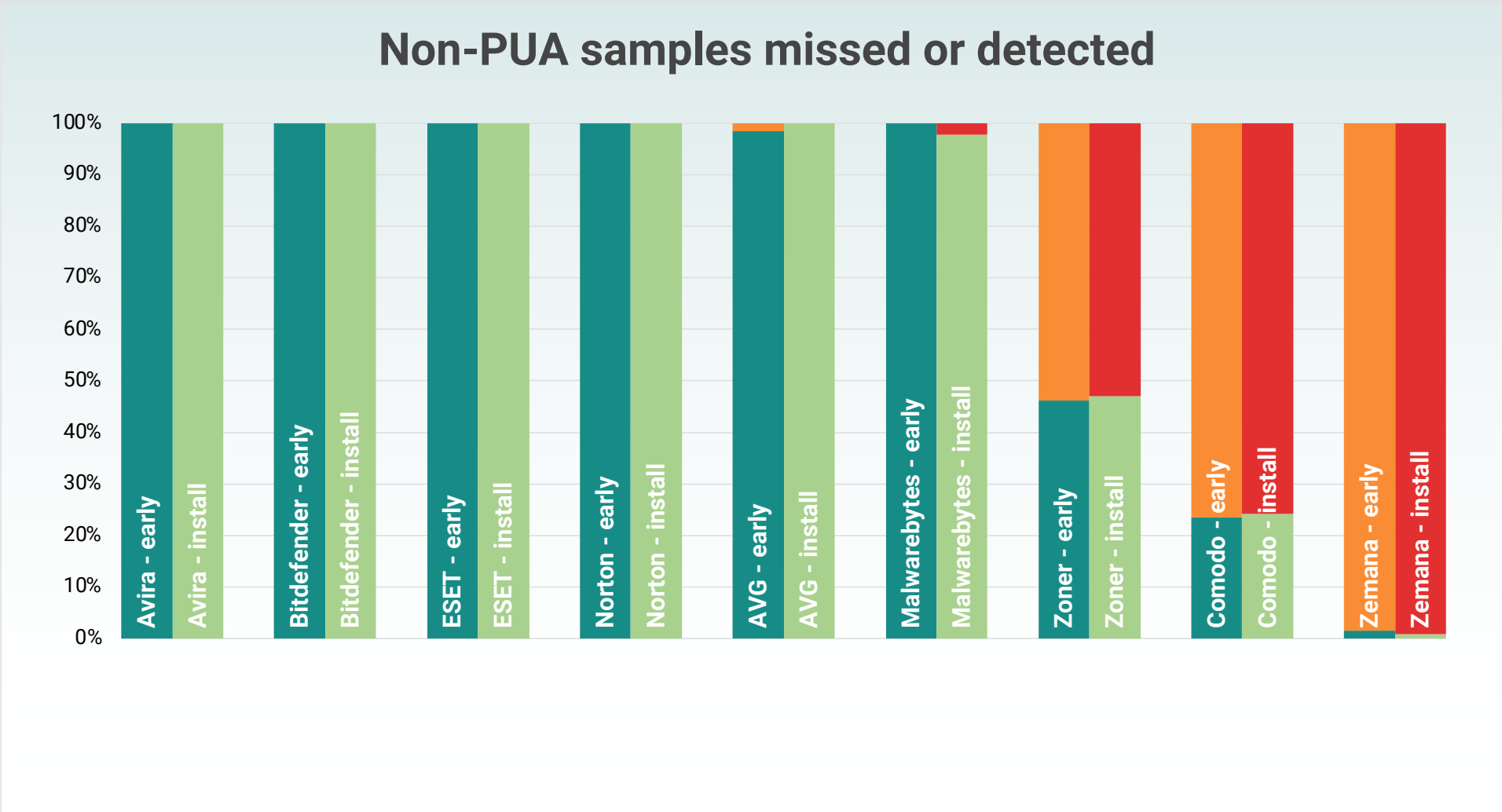


FIGURE 4. SUMMARY, NON-PUA SAMPLES

Non-PUA sum summary								
Short name	Early blocked	Early blocked	Early missed	Early missed	Install blocked	Install blocked	Install missed	Install missed
Avira	132	100,0%	0	0,0%	132	100,0%	0	0,0%
Bitdefender	132	100,0%	0	0,0%	132	100,0%	0	0,0%
ESET	132	100,0%	0	0,0%	132	100,0%	0	0,0%
Norton	132	100,0%	0	0,0%	132	100,0%	0	0,0%
AVG	130	98,5%	2	1,5%	132	100,0%	0	0,0%
Malwarebytes	132	100,0%	0	0,0%	129	97,7%	3	2,3%
Zoner	61	46,2%	71	53,8%	62	47,0%	70	53,0%
Comodo	31	23,5%	101	76,5%	32	24,2%	100	75,8%
Zemana	2	1,5%	130	98,5%	1	0,8%	131	99,2%

TABLE 3. RESULTS, NON-PUA SAMPLES

PUA Detection

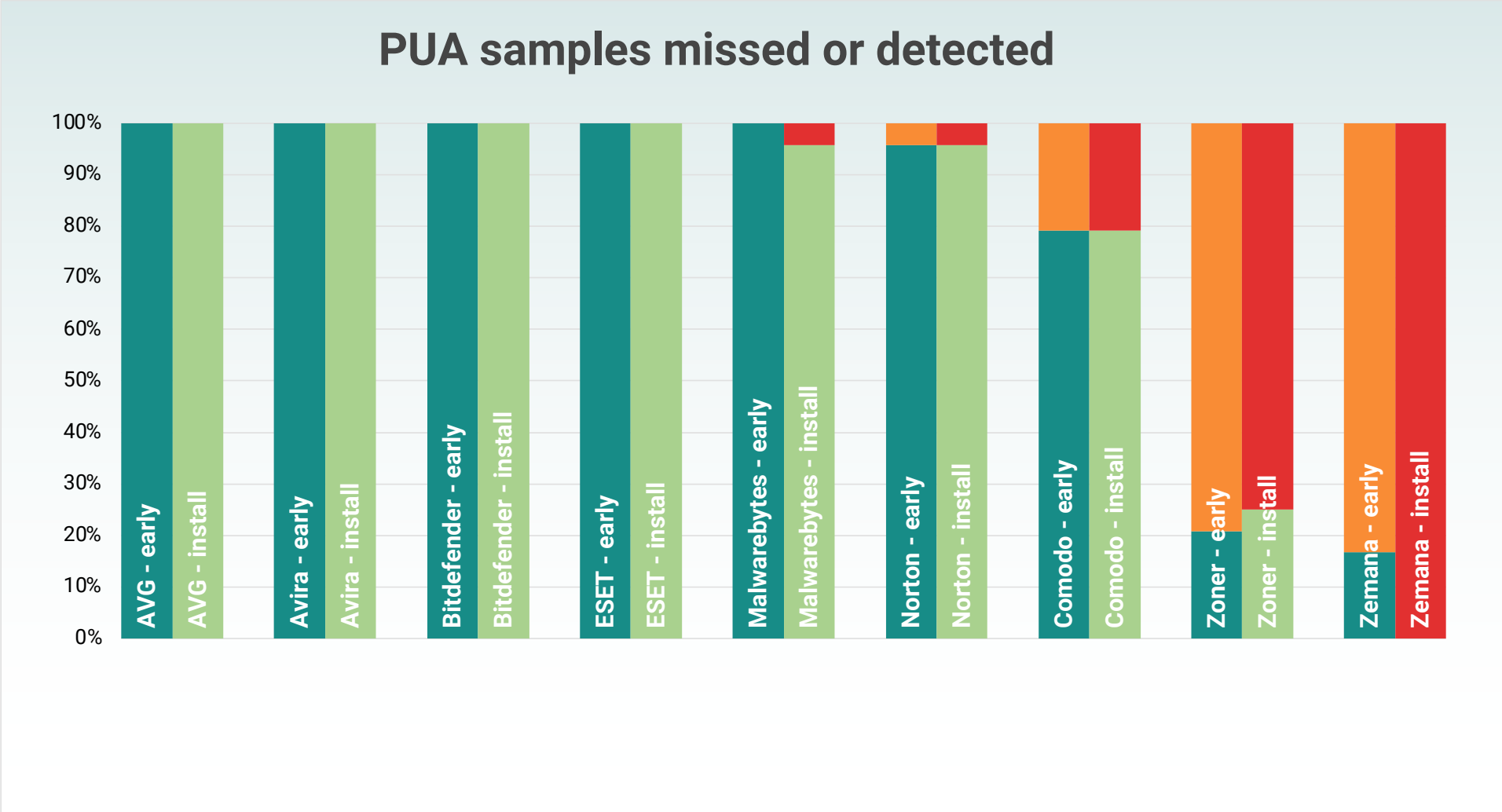


FIGURE 5. SUMMARY, PUA SAMPLES

PUA summary								
Short name	Early blocked	Early blocked	Early missed	Early missed	Install blocked	Install blocked	Install missed	Install missed
AVG	24	100,0%	0	0,0%	24	100,0%	0	0,0%
Avira	24	100,0%	0	0,0%	24	100,0%	0	0,0%
Bitdefender	24	100,0%	0	0,0%	24	100,0%	0	0,0%
ESET	24	100,0%	0	0,0%	24	100,0%	0	0,0%
Malwarebytes	24	100,0%	0	0,0%	23	95,8%	1	4,2%
Norton	23	95,8%	1	4,2%	23	95,8%	1	4,2%
Comodo	19	79,2%	5	20,8%	19	79,2%	5	20,8%
Zoner	5	20,8%	19	79,2%	6	25,0%	18	75,0%
Zemana	4	16,7%	20	83,3%	0	0,0%	24	100,0%

TABLE 4. PUA SAMPLES BLOCKED OR MISSED

Trojan Detection

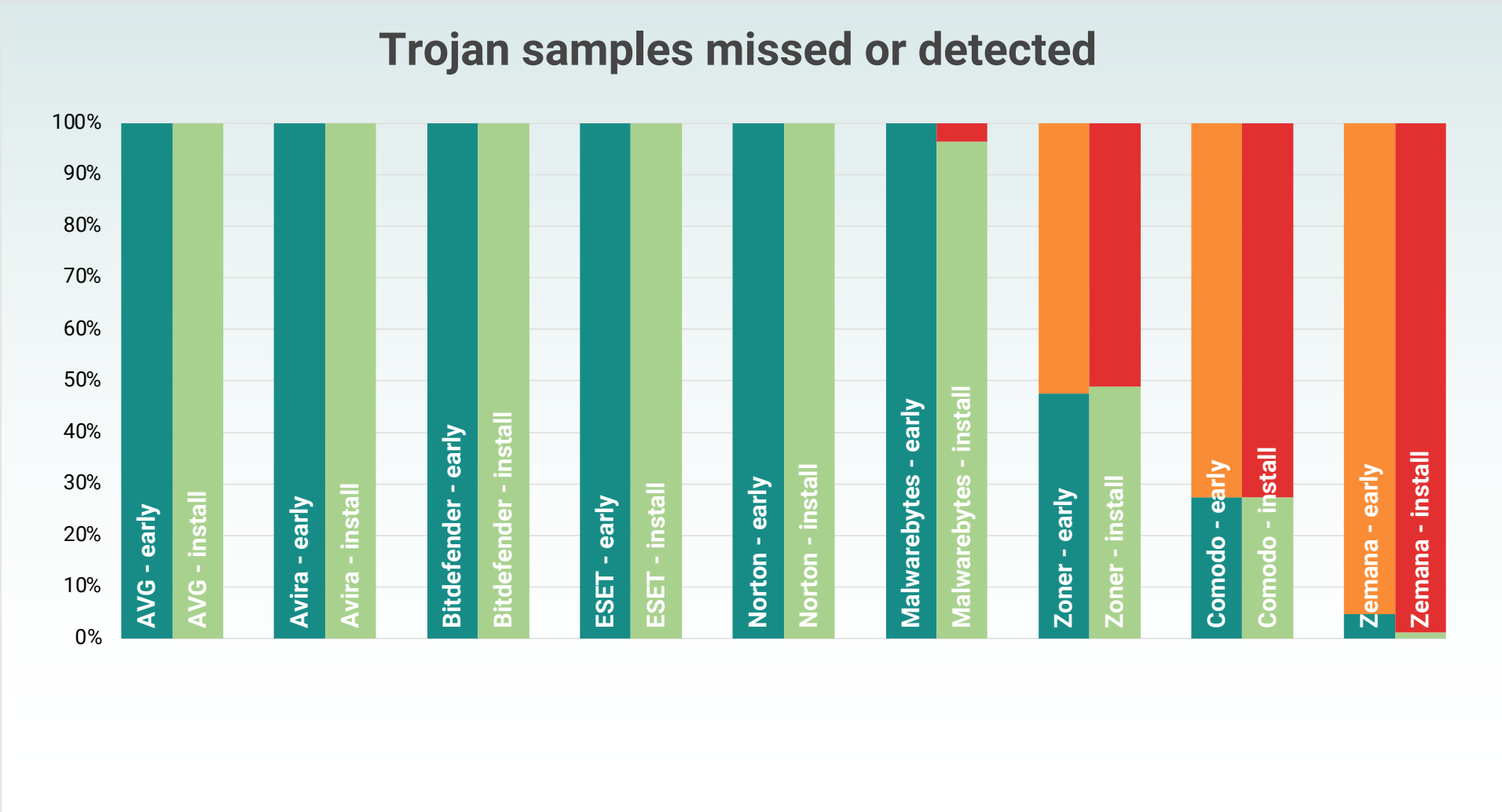


FIGURE 6. SUMMARY, TROJAN SAMPLES

Trojan summary								
Short name	Early blocked	Early blocked	Early missed	Early missed	Install blocked	Install blocked	Install missed	Install missed
AVG	84	100,0%	0	0,0%	84	100,0%	0	0,0%
Avira	84	100,0%	0	0,0%	84	100,0%	0	0,0%
Bitdefender	84	100,0%	0	0,0%	84	100,0%	0	0,0%
ESET	84	100,0%	0	0,0%	84	100,0%	0	0,0%
Norton	84	100,0%	0	0,0%	84	100,0%	0	0,0%
Malwarebytes	84	100,0%	0	0,0%	81	96,4%	3	3,6%
Zoner	40	47,6%	44	52,4%	41	48,8%	43	51,2%
Comodo	23	27,4%	61	72,6%	23	27,4%	61	72,6%
Zemana	4	4,8%	80	95,2%	1	1,2%	83	98,8%

TABLE 5. RESULTS, TROJAN SAMPLES

Banking Detection

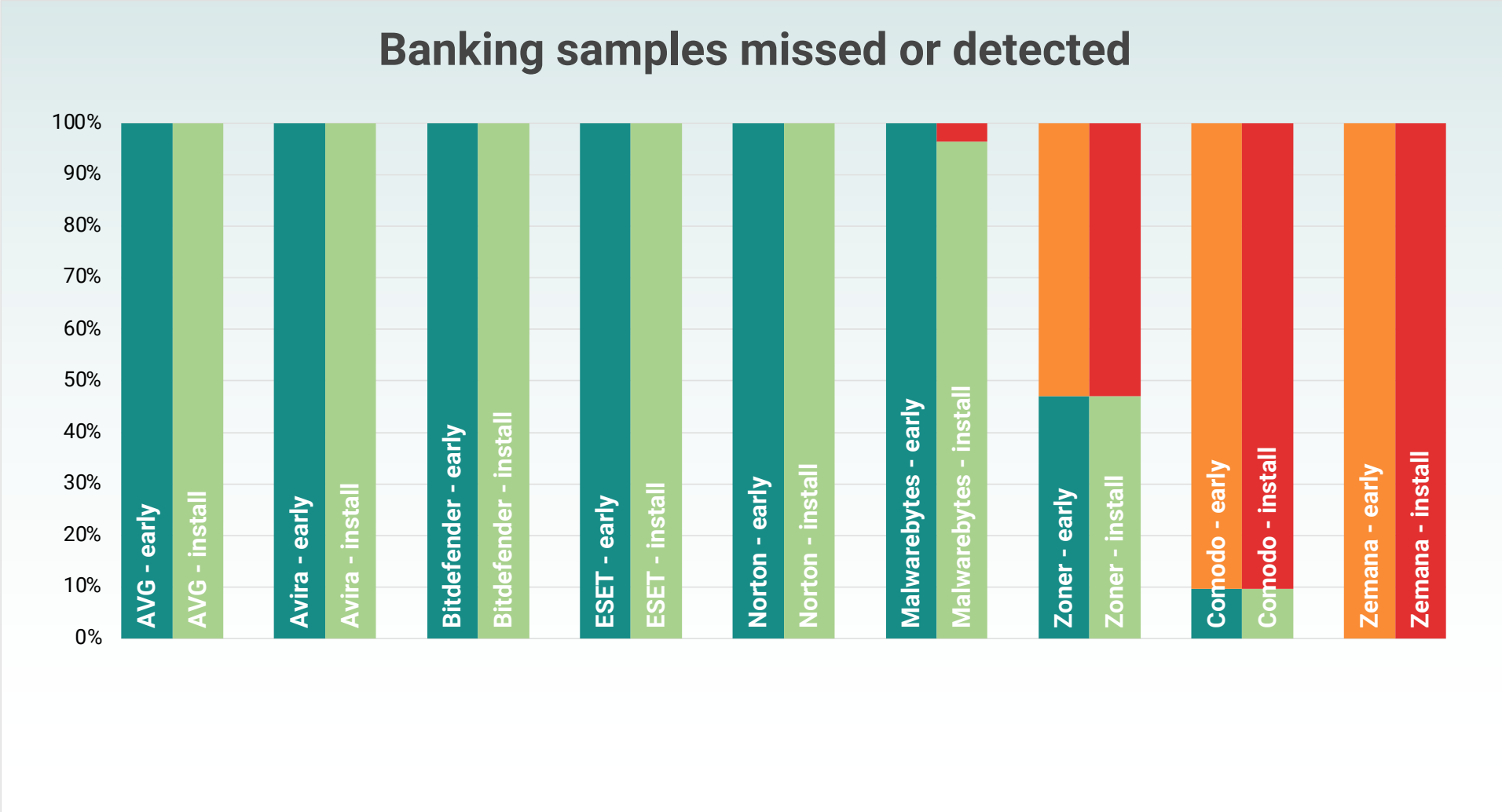


FIGURE 7. SUMMARY, BANKING SAMPLES

Banking summary								
Short name	Early blocked	Early blocked	Early missed	Early missed	Install blocked	Install blocked	Install missed	Install missed
AVG	83	100,0%	0	0,0%	83	100,0%	0	0,0%
Avira	83	100,0%	0	0,0%	83	100,0%	0	0,0%
Bitdefender	83	100,0%	0	0,0%	83	100,0%	0	0,0%
ESET	83	100,0%	0	0,0%	83	100,0%	0	0,0%
Norton	83	100,0%	0	0,0%	83	100,0%	0	0,0%
Malwarebytes	83	100,0%	0	0,0%	80	96,4%	3	3,6%
Zoner	39	47,0%	44	53,0%	39	47,0%	44	53,0%
Comodo	8	9,6%	75	90,4%	8	9,6%	75	90,4%
Zemana	0	0,0%	83	100,0%	0	0,0%	83	100,0%

TABLE 6. RESULTS, BANKING SAMPLES

SMS Detection

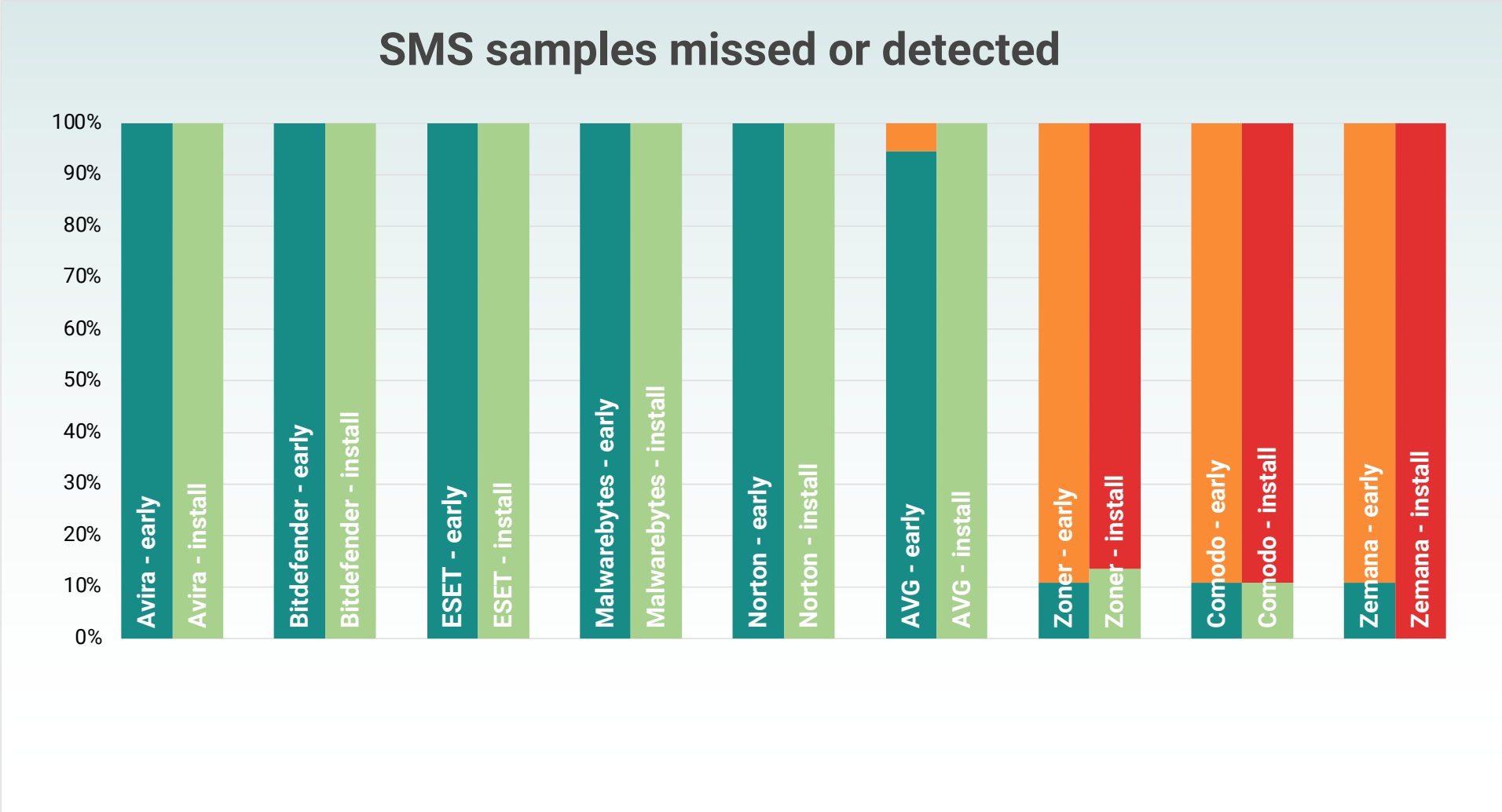


FIGURE 8. SUMMARY, SMS SAMPLES

SMS summary								
Short name	Early blocked	Early blocked	Early missed	Early missed	Install blocked	Install blocked	Install missed	Install missed
Avira	37	100,0%	0	0,0%	37	100,0%	0	0,0%
Bitdefender	37	100,0%	0	0,0%	37	100,0%	0	0,0%
ESET	37	100,0%	0	0,0%	37	100,0%	0	0,0%
Malwarebytes	37	100,0%	0	0,0%	37	100,0%	0	0,0%
Norton	37	100,0%	0	0,0%	37	100,0%	0	0,0%
AVG	35	94,6%	2	5,4%	37	100,0%	0	0,0%
Zoner	4	10,8%	33	89,2%	5	13,5%	32	86,5%
Comodo	4	10,8%	33	89,2%	4	10,8%	33	89,2%
Zemana	4	10,8%	33	89,2%	0	0,0%	37	100,0%

TABLE 7. RESULTS, SMS SAMPLES

Spyware Detection

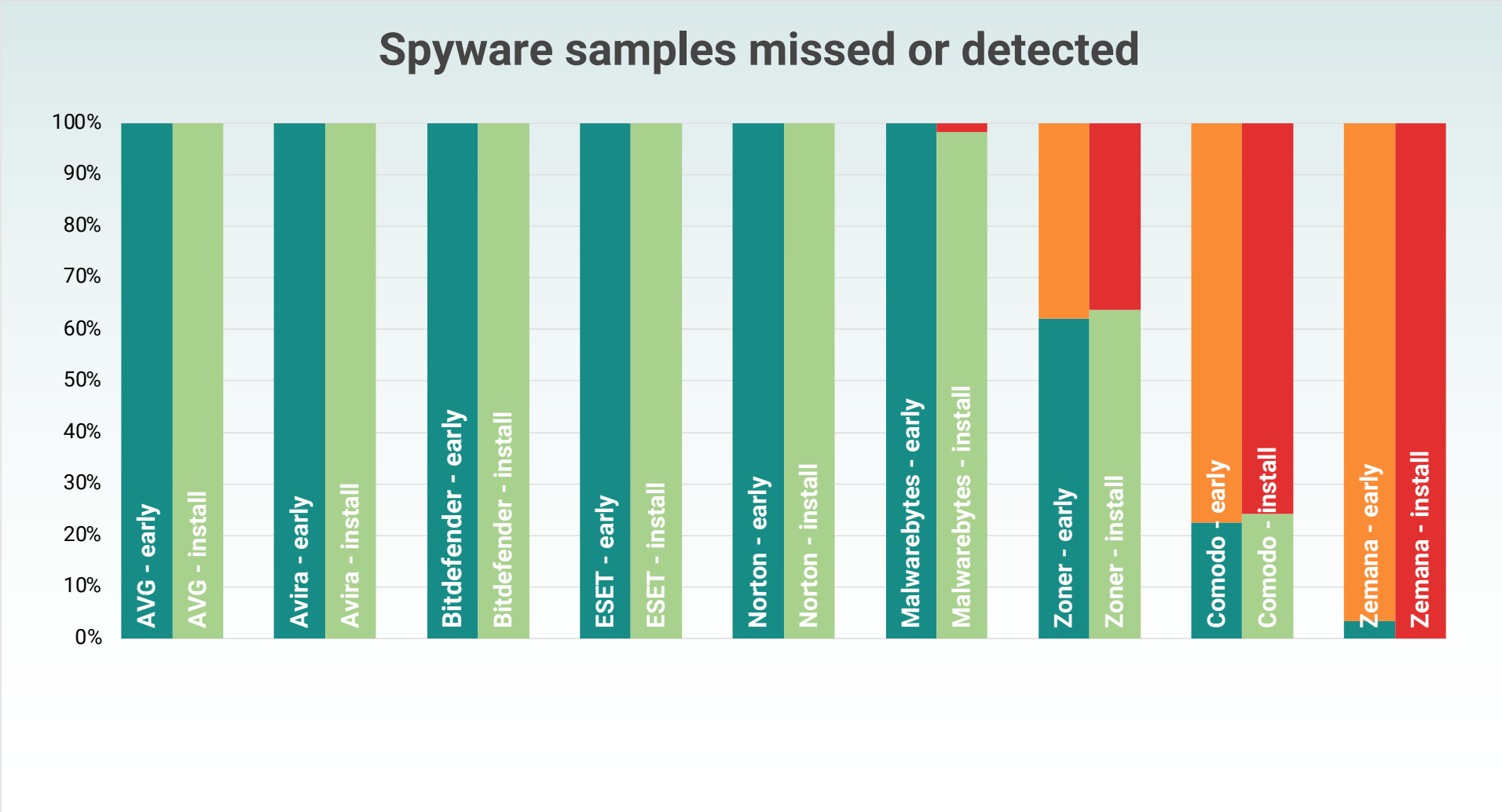


FIGURE 9. SUMMARY, SPYWARE SAMPLES

Spyware summary								
Short name	Early blocked	Early blocked	Early missed	Early missed	Install blocked	Install blocked	Install missed	Install missed
AVG	58	100,0%	0	0,0%	58	100,0%	0	0,0%
Avira	58	100,0%	0	0,0%	58	100,0%	0	0,0%
Bitdefender	58	100,0%	0	0,0%	58	100,0%	0	0,0%
ESET	58	100,0%	0	0,0%	58	100,0%	0	0,0%
Norton	58	100,0%	0	0,0%	58	100,0%	0	0,0%
Malwarebytes	58	100,0%	0	0,0%	57	98,3%	1	1,7%
Zoner	36	62,1%	22	37,9%	37	63,8%	21	36,2%
Comodo	13	22,4%	45	77,6%	14	24,1%	44	75,9%
Zemana	2	3,4%	56	96,6%	0	0,0%	58	100,0%

TABLE 8. RESULTS, SPYWARE SAMPLES

Simulators

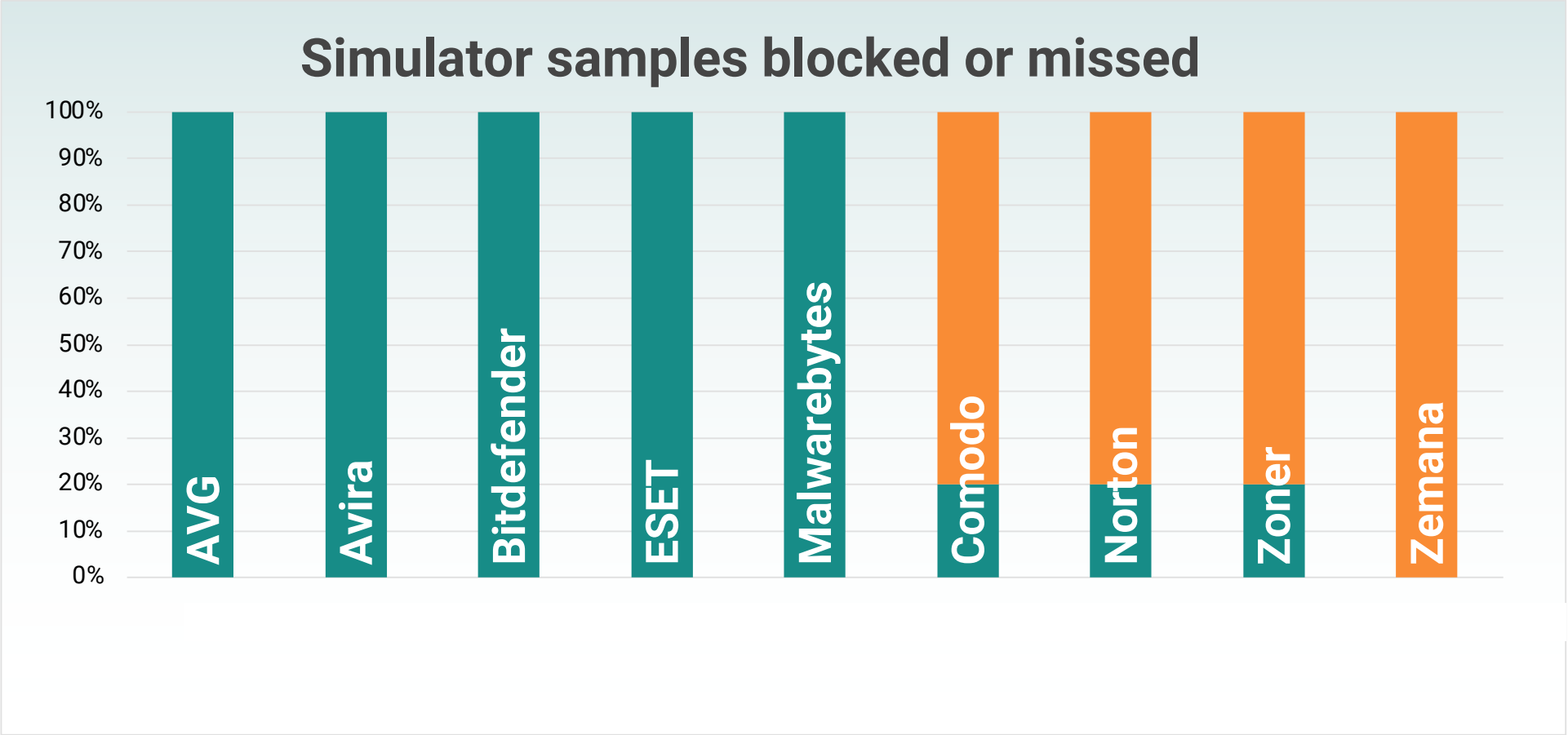


FIGURE 10. SUMMARY, SIMULATOR SAMPLES

Simulator summary				
Full name	Blocked	Blocked	Missed	Missed
AVG	5	100%	0	0%
Avira	5	100%	0	0%
Bitdefender	5	100%	0	0%
ESET	5	100%	0	0%
Malwarebytes	5	100%	0	0%
Comodo	1	20%	4	80%
Norton	1	20%	4	80%
Zoner	1	20%	4	80%
Zemana	0	0%	5	100%

TABLE 9. RESULTS, SIMULATOR SAMPLES

False positive tests

All test participants achieved 100% results in this category.

Summary

The following AV engines reached a 99% detection rate in a non-PUA sample set, therefore, they have been awarded with the MRG Effitas Certificate.

- AVG
- Avira
- Bitdefender
- ESET
- Norton
- Malwarebytes

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.

Vendor reputation and extra services

As of Q2 2021, the majority of the well-established vendors reached a perfect or near-perfect score in the in-the-wild categories. From a user perspective, this is all good news as several viable options are provided, some even without a subscription fee. In the future, we expect that the extra features (VPN, family locator, connections with desktop AV licenses etc.) will have a significant effect on user choice in the future.

As user consciousness evolves, more and more emphasis will be put on early detection, as a successful AV needs to provide functionality to scan the SD card for malicious content.

'AV as another app'

Testing led us to the conclusions that detection for many 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, as "AV is another app". As a result, having already started the freshly installed sample, it is quite hard to get rid of some samples in the in-the-wild test set, making a timely and properly displayed detection an absolute must for AV engines.

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.

Furthermore, the general idea of providing Early scan features, is a significant feature for the security conscious. As it provides an opportunity to scan an APK file – prior to installation – it makes it possible to detect malicious applications, prior to being installed and sitting only one tap away from causing all sorts of havoc. In our opinion, Early scan features should be more prominent in all Android based AVs (in our recent tests, we almost always found applications, not providing this feature).

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 simulator samples.

Our test lab has been reached out to on several occasions with claims that the simulators we utilise, do not present a lifelike challenge for an AV engine. However, field reports show the presented scenario – when an adversary patches an existing Android application to perform hidden spying activity – is well known and have been utilised for almost a decade[7]. The most famous of campaigns with this approach, is still the Dark Caracal APT[8], with an excellent analysis by Lookout[9].

Detection notification

During the history of 360 Android tests, we noticed that there are significant differences in terms of user notification. When it comes to a successful detection, efficient and clear user notification is an essential part of both the efficacy of the AV application and the overall user experience.

The tested AVs choose one of the following approaches.

1. A separate activity is launched, usually with a bright red background and a couple of lines describing the nature of the threat, presented by the freshly installed application.
2. The Android notification subsystem is utilised to issue an important notification, usually displayed in the status bar.

Both approaches have their merits, namely a separate activity is harder to dismiss or overlook, and the second one being more streamlined with the overall Android experience. However, Android provides a lot of options for users to customize notifications, therefore it is possible for the notification to get lost in the clutter, therefore many of the tested apps opt for the first approach.

[7] https://threatvector.cylance.com/en_us/home/mobile-malware-and-apt-espionage-prolific-pervasive-and-cross-platform.html

[8] https://en.wikipedia.org/wiki/Dark_Caracal

[9] https://www.lookout.com/info/ds-dark-caracal-ty

Furthermore, applying too many or overly frequent notification in the notification bar might result in important notifications getting lost in the noise. For instance, an AV might wish to inform the user that some kind of background activity is taking place, in order to make it apparent that the device is provided with protection – however, should the user be trained to have AV notifications all the time, might just form an automatic dismissal reaction. As always, this is a matter of finding an important balance.

As for the wording, the overall design of the displayed notification and the consequent user choice description, there is a significant room for improvement in many apps. The Android way is to communicate as much information to the user as possible, just enough so that they can make a responsible decision – however, a responsible user has to read and process the text displayed on the screen, which presents a significant mental load, especially for the not tech savvy. As a result, a well-designed GUI can make all the difference for the everyday user, when it comes to making users' choices more responsible (and consequently, their devices more secure).

MRG Effitas

EFFICACY ASSESSMENT & ASSURANCE