Malware in the Mobile Device Android Environment

Diana Hintea  
*Coventry University, School of Computing, Electronics and Maths, diana.hintea@coventry.ac.uk*

Robert Bird  
*Coventry University, School of Computing, Electronics and Maths, robert.bird@coventry.ac.uk*

Andrew Walker  
*BAE Systems Applied Intelligence Guildford, walker65@uni.coventry.ac.uk*

Follow this and additional works at: [https://commons.erau.edu/adfsl](https://commons.erau.edu/adfsl)

Part of the Aviation Safety and Security Commons, Computer Law Commons, Defense and Security Studies Commons, Forensic Science and Technology Commons, Information Security Commons, National Security Law Commons, OS and Networks Commons, Other Computer Sciences Commons, and the Social Control, Law, Crime, and Deviance Commons

Scholarly Commons Citation

[https://commons.erau.edu/adfsl/2016/wednesday/6](https://commons.erau.edu/adfsl/2016/wednesday/6)

(c)ADFSL

This Peer Reviewed Paper is brought to you for free and open access by the Conferences at Scholarly Commons. It has been accepted for inclusion in Annual ADFSL Conference on Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.
MALWARE IN THE MOBILE DEVICE
ANDROID ENVIRONMENT

Diana Hintea
Coventry University
School of Computing, Electronics and Maths
Coventry, CV1 2JH
diana.hintea@coventry.ac.uk

Robert Bird
Coventry University
School of Computing, Electronics and Maths
Coventry, CV1 2JH
robert.bird@coventry.ac.uk

Andrew Walker
BAE Systems Applied Intelligence
Guildford, GU2 7RQ
walker65@uni.coventry.ac.uk

ABSTRACT
Consequent to the world wide increase of smartphone use, the incidence of malware developed to exploit smartphone operating systems has exponentially expanded. Android has become the main target to exploit due to having the largest install base amongst the smartphone operating systems and owing to the open access nature in which application installations are permitted. Many Android users are unaware of the risks associated with a malware infection and to what level current malware scanners protect them. This paper tests how efficient the currently available malware scanners are. To achieve this, ten representative Android security products were selected and tested against a set of 5,560 known and categorized Android malware samples. The tests were carried out using a digital-forensically rigorous testing framework and methodology, which ensures the scientific validity of the results. The detection rates of the tested malware scanners varied widely with half unable to detect any samples at all during initial testing. The malware scanners that were able to detect the samples scored highly with the top four between 97-99% and a fifth scanner scoring 87%. The results emphasise the need for more complex detection mechanisms and protections in future versions of Android and the next generation of malware scanners.

Keywords: malware, mobile forensics, Android

1. INTRODUCTION
The Android operating system and its users are a significant target for malware developers; however, many Android users are unaware of what malware is and the threats associated with it. The first objective of this paper is to establish whether Android is more susceptible to malware than other mobile operating systems and if so, why. Android users that are more security-conscious will often download and use malware scanners in the expectation
that it will protect them from malware, but
the effectiveness of these scanners is yet to be
investigated. The second part of this paper
answers this concern by testing a
representative set of malware scanners against
known malware, making use of both static
analysis and virtual devices.

The paper is structured as follows: Section 2
reviews the associated literature review,
while Section 3 describes the research
methodology. Section 4 presents the setup
procedure, while Section 5 provides details on
the testing methodology. The results obtained
through testing are given in Section 6. Finally,
Section 7 concludes the paper.

2. LITERATURE REVIEW
This section describes the related literature in
three different areas: i) Android operating
system features and malware, ii) the threat of
Android malware, and iii) the effectiveness of
malware scanners in Android.

2.1 Malware and the Android
operating system

Android is a mobile operating system based on
the Linux kernel and developed in part by
Google. The operating system was launched in
September 2008 (Morrill, 2008) and has since
grown rapidly. As of 2012, Android has the
largest installed base of any mobile platform,
powers hundreds of millions of mobile devices
and has more than 1 million new device
activations each day (Android Open Source
Project, 2012).

Having the largest market share makes
Android a prime target for malware developers
as they seek to spread infection promiscuously
and to maximise their impact. Moreover,
Motive Security Labs identified in their H2
2014 report what they consider to be key
reasons why malware continues to be a
significant problem in Android, whilst also
performing a comparison with other mobile
platforms. The reasons mentioned are the
following:

1. Android apps can be downloaded from
third-party app stores and web sites.
2. There is no control of the digital
certificates used to sign Android apps.
3. Android apps are usually self-signed
and can’t be traced to the developer.
4. It is easy to hijack an Android app,
inject code into it and re-sign it.

In terms of other smartphones, such as
iPhone, Blackberry or Windows Mobile, they
make up less than 1% of the infections
observed. The iPhone and Blackberry have a
more controlled app distribution environment
and are thus less of a target (Motive Security
Labs, 2014). Motive Security noted that
Android phones and tablets are responsible for
around 50% of the malware infections
observed. These results are based on analysis
of all platforms, including both desktop and
mobile operating systems.

2.2 The threat of Android
malware

To understand why malware is so prolific in
Android, it is important to understand what it
seeks to do and the unique opportunities,
exploiting a mobile operating system present to
malware developers. Zhou and Jiang (2012)
discuss the threat of Android malware. In the
paper, they focus on the Android platform and
aim to systematize or characterize existing
Android malware. In order to achieve this,
they collected 1,200 malware samples that
cover the majority of existing Android malware
“families.” Although this appears to be a small
number of samples by statistical standards, at
the time these samples formed the first large
collection of Android malware sample and
represented the majority of existing Android
malware.
Their findings determine that 86% of the samples are repackaged versions of legitimate applications with malicious payloads and can therefore be broadly categorised as Trojans. Android users should be aware of this statistic and should avoid downloading applications from third parties even if they appear to be genuine applications. Another finding of this report is that 36.7% of the collected malware samples leverage root-level exploits. This result shows how important it is for Android users to keep their operating systems “patched” or up to date. Older versions of Android are potentially vulnerable to root-level exploits that have since been patched.

The report found that 45.3% of the samples have the built-in support of transmitting background short messages (to premium-rate numbers) or initiating phone calls without user awareness. This category of malware could cause financial penalty to a user if they are infected as they could incur a large phone bill that they are required to pay to their provider. This result also enforces the previous observation that the Android operating system should be patched regularly. This is because in Android 4.2 onwards, a security mechanism was introduced whereby the operating system will alert the user before sending a premium rate SMS message (Eitzen, 2012). The final finding of this report is that 51.1% of the malware samples are harvesting users’ information, including user accounts and short messages stored on the phones (Zhou and Jiang, 2012). The information gathered by this type of malware could be personally or commercially compromising and could lead to crimes such as identify theft for the individual, or intellectual property and theft of commercial secrets for a business.

This paper identifies the key threats of Android malware - information theft and fraud through the misuse of premium rate texts and calls. Mobile operating systems, such as Android, store significant amounts of personal and business information that is potentially valuable to malware developers. Secondly, it finds that mobile operating systems frequently run on devices that contain SIM cards, making it possible for malware developers to obtain money directly through fraudulent phone charges.

2.3 The effectiveness of malware scanners in Android

A recent paper that discusses the effectiveness of malware scanners in Android is: “DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket” (Arp et al. 2014); hence-forth referred to as the DREBIN paper. In the DREBIN paper ten “anti-virus” scanners were tested “(Avira, AVG, Bit-Defender, ClamAV, ESET, F-Secure, Kaspersky, McAfee, Panda, Sophos)” against a set of 5,560 malware samples; the same malware set used in this research. The paper discusses the creation of an Android application called DREBIN which is a lightweight method for detection of Android malware that enables identifying malicious applications directly on the smartphone.

The application was tested on Android and detects 94% of the malware; however, the other “anti-virus” products’ detections rates were checked by uploading the samples to the VirusTotal website (VirusTotal, 2015) and recording the result returned by the service. This disparity of methods renders comparing results accurately impossible. This is because VirusTotal does not differentiate between different products from the same vendor. For example, the AVG product for Windows desktop and Android may work differently and report a different detection rate, but VirusTotal only outputs a detection result for AVG as a single result. Further, VirusTotal does not recommend that its service is used for this purpose. Under the heading “BAD IDEA:
VirusTotal for antivirus/URL scanner testing” on their website’s about page they state:

“At VirusTotal we are tired of repeating that the service was not designed as a tool to perform antivirus comparative analyses…” Those who use VirusTotal to perform antivirus comparative analyses should be aware that they are introducing many implicit errors in their methodology, the most obvious being:

1. VirusTotal’s antivirus engines are command line versions, so depending on the product, they will not behave exactly the same as the desktop versions: for instance, desktop solutions may use techniques based on behavioural analysis and count with personal firewalls that may decrease entry points and mitigate propagation, etc.

2. Some of the solutions included in VirusTotal are parametrized (in coherence with the developer company’s desire) with a different heuristic/aggressiveness level than the official end-user default configuration (VirusTotal, 2015).

For the above stated reasons, the authors decided to test the detection rates of malware scanners available on the Android operating system using the same malware samples. Using this method will give an accurate representation of the detection rates users would experience if they were to come into contact with that malware on their Android device.

3. RESEARCH METHODOLOGY

The key questions this research addresses are the following:

1. If Android is a particular target of malware, are there any protections against malware within the operating system itself?
2. What are the threats it poses? This section of the question seeks to understand the goals of malware created for Android and the threat it poses to users, their devices and data.
3. How effective are malware scanners in detecting and addressing those threats? This section of the question focuses on malware defense and the malware products created by vendors to protect against malware.

The first section is best addressed by using the literature review. Sources that discuss the evolution of the Android operating system and malware created were used. The authors also believe that a literature review is most appropriate for the second section of the question. Sources that analyse the goal of Android malware and the threat it poses will be used. For the third section the authors chose to use a testing methodology. We test a known set of malware samples against ten representative Android malware scanning products using static analysis. The results will be used to determine a detection rate for each product. The testing methodology was chosen for the following reasons:

1. The Android operating system is constantly updated. It is difficult to find sources that test malware in the latest versions of Android.
2. Malware scanners and their definitions are continuously updated. New malware scanners that may not have previously been tested against a dataset may have become available since the source published, or a product may have improved its detection rate.
3. Any source that carried out testing previously is quickly out of date as definitions in malware scanners are updated; in most cases daily. By
carrying out our own testing we will have an accurate, up-to-date snapshot of the latest products and their definitions.

4. Using data from third party sources such as VirusTotal may not give accurate results.

4. SETUP PROCEDURE

4.1 Selection of malware samples

The authors used the DREBIN malware dataset (Arp et al., 2014) for testing. This is a well-defined and ample database of malware samples. The DREBIN dataset contains 5,550 malware samples collected in the period from August 2010 to October 2012. Each sample was uploaded to the VirusTotal (VirusTotal 2015) service and checked against the ten most common malware scanners, namely AntiVir, AVG, BitDefender, ClamAV, ESET, F-Secure, Kaspersky, McAfee, Panda, and Sophos. Applications were categorised as malicious only if they were detected as so by at least two of the products. The dataset also includes all 1260 samples from the Android Malware Genome Project (Zhou and Jiang, 2012). Additionally, any samples that were identified as adware were excluded from the dataset. The dataset contains malware from 178 different identified malware families with 81.15% of the malware belonging to the top twenty families in the dataset.

Table 1
The Android Application Package install files.

<table>
<thead>
<tr>
<th>Reference</th>
<th>APK Name</th>
<th>Version</th>
<th>APK MD5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1CM</td>
<td>com.cleanmaster.security.apk</td>
<td>2.4.9</td>
<td>114a64167a7e59edc94d708ac2795c01</td>
</tr>
<tr>
<td>2Q360</td>
<td>com.qihoo.security-67.apk</td>
<td>3.2.1</td>
<td>fa8aeb6d65979a8cc13db8d7f8fe</td>
</tr>
<tr>
<td>3AVG</td>
<td>com.antivirus.apk</td>
<td>4.3</td>
<td>75d2e1597a6d056f6d297774f5</td>
</tr>
<tr>
<td>4McAfee</td>
<td>com.wsandroid.suite.apk</td>
<td>4.4.0.392</td>
<td>c44c8dc58b565a4935bab0f17370b7</td>
</tr>
<tr>
<td>5AVAST</td>
<td>com.avast.android.mobilesecurity.apk</td>
<td>4.0.7800</td>
<td>b9a8f4a1ca3fa4414f4b192bb7090e5</td>
</tr>
<tr>
<td>6Norton</td>
<td>com.symantec.mobilesecurity</td>
<td>3.10.0.2360</td>
<td>ca3f66d072f6e71598b32a81f8ca8d2</td>
</tr>
<tr>
<td>7Lookout</td>
<td>com.lookout</td>
<td>9.15-26a3b5f</td>
<td>9000e7207cd4b444af1b84ca2b60e</td>
</tr>
<tr>
<td>8NQ</td>
<td>com.zrgiu.antivirus</td>
<td>7.3.12.02</td>
<td>7820cb011a8a99b76166557b8e32ed3</td>
</tr>
<tr>
<td>9Malwarebytes</td>
<td>org.malwarebytes.antimalware</td>
<td>1.05.1.1000</td>
<td>8740049b911021d6e42f010a1102aa6</td>
</tr>
<tr>
<td>10Kaspersky</td>
<td>com.kms.free-84.apk</td>
<td>11.8.4.474</td>
<td>7c7eef4f2a4ed445d97f6d21c5319559</td>
</tr>
</tbody>
</table>

The authors selected the DREBIN dataset as it contains a large number of samples across a substantial number of malware families, facilitating a comprehensive comparative test. The dataset was collected over a long period of time and is therefore likely to contain samples targeting devices running both newer and older versions of Android. The DREBIN paper itself states that “To the best of our knowledge, this is one of the largest malware datasets that has been used to evaluate a malware detection method on Android” (Arp et al., 2014). When creating the malware set, a number of steps have been taken to make the samples more reliable by requiring the samples to be categorised as malicious by more than one of the tested antivirus products and by excluding adware as it exists in a “twilight zone between malware and benign functionality” (Arp et al., 2104).
4.2 Selection of malware scanners

The authors selected applications to test from the Google Play Store based on the following criteria:

- Any application that identifies itself as an antimalware or antivirus product and appears in Google’s Play store as one of their “Top Apps.” Google lists 540 such applications.
- Any application that identifies itself as an antimalware or antivirus product and appears in Google’s Play Store as one of the top applications for a category, for example “Top Tools Apps.”

This selection process was limited to the top 100 applications in each category. The Google Play store contains the following categories: Books & Reference, Business, Comics, Communication, Education, Entertainment, Finance, Health & Fitness, Libraries & Demo, Lifestyle, Live Wallpaper, Media & Video, Medical, Music & Audio, News & Magazines, Personalisation, Photography, Productivity, Shopping, Social, Sports, Tools, Transport, Travel & Local, Weather, and Widgets.

![Android Virtual Device (AVD) Manager](Figure 2 - Creating Android Virtual Devices)
Table 2

<table>
<thead>
<tr>
<th>MD5 and SHA1 hashes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MD5</strong></td>
</tr>
<tr>
<td>8GB SD – No malware</td>
</tr>
<tr>
<td>8GB SD - With DREBIN malware Samples</td>
</tr>
</tbody>
</table>

Table 3

<table>
<thead>
<tr>
<th>Malware applications selected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reference</strong></td>
</tr>
<tr>
<td>1CM</td>
</tr>
<tr>
<td>2Q360</td>
</tr>
<tr>
<td>3AVG</td>
</tr>
<tr>
<td>4McAfee</td>
</tr>
<tr>
<td>5AVAST</td>
</tr>
<tr>
<td>6Norton</td>
</tr>
<tr>
<td>7Lookout</td>
</tr>
<tr>
<td>8NQ</td>
</tr>
<tr>
<td>9Malwarebytes</td>
</tr>
<tr>
<td>10Kaspersky</td>
</tr>
</tbody>
</table>

The rationale behind this selection process is that users will browse the “Top Apps” when selecting a malware scanning product. Another method would be to use search terms within the Play Store, such as “antivirus” and “malware”. This is problematic due to the naming conventions of applications, key words and the search algorithm used by the Play Store. This would also introduce bias, as key words and terms we associate with malware scanners are likely to differ from that of the average user. The applications listed in Table 3 were selected.

The APK (Android Application Package) install files were then downloaded for the selected applications from the Google Play Store.

4.3 Using AVD (Android Virtual Device)

For testing, the authors decided to use virtualised Android phones by using AVD (Android Virtual Device) which is included in the Android SDK. A virtual Android phone was used for testing each malware scanner. Each phone was configured in AVD as illustrated in Figure 2.

Each virtual device is named with the reference of the scanner to be tested on that device. The Nexus 5 device template was chosen and the device was set to run the latest version of Android (5.1.1 13/04/2015). For the CPU architecture ARM was chosen so as to replicate the architecture found on the physical Nexus 5 phone. Cameras for the device were emulated, 2048MB of RAM (default) and 200MB of internal storage (default) were allocated. The virtual SD card (Section 6.4) containing the malware samples for each device is attached.

4.4 Virtual SD Card
Once the decision to use virtual devices for testing was made, the authors decided to create an SD card image for each device using the Android SDK tool “mksdcard” in order to load the malware samples onto each virtual device. The authors created a blank 8GB SD card using the command: “mksdcard.exe 8G 8GBSD” and mounted the image in Ubuntu and copied all 5560 malware samples onto the SD card image, followed by un-mounting the image. The created image served as the master file for all the virtual device’s SD cards and was not used in testing. An SD card image was made for each virtual device and named after its reference number used in the test. A virtual SD card was used, as it was easy to verify all the samples to be loaded by checking the hash of the SD card image. Furthermore, using a virtual SD card is a lot faster than booting the phone and copying each malware sample individually.

5. TESTING
The following steps were followed to test each malware scanner. It is important that testing occurs in a forensically-sound environment so that a third party performing the same tests would achieve the same results. When carrying out the testing, the authors adhered to the UK ACPO (Association of Chief Police Officers) principles (ACPO, 2012) which provides guidelines on the handling of digital evidence.

Although these principles are aimed at law enforcement and admissibility of evidence in court, they serve as a solid framework for this type of testing. ACPO Principle 3 will be followed and states: “An audit trail or other record of all processes applied to digital evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result” (ACPO 2012).

Steps to reproduce:

1. Verify the SD card image to be tested. The MD5 and SHA1 hash should match that of the original SD card image.
2. Configure the Android Virtual device and attach the applicable virtual SD card containing the malware samples.
3. Select the device in AVD manager and select “Start” to power on the device.
4. Locate “adb.exe,” installed as part of the Android SDK.
5. At the command line, call “adb.exe shell.” This will provide shell access to the device.
   a. At the shell, navigate to the mounted SD card using “cd /storage/sdcard.”
   b. Use the ls command to list the mounted files “ls –la.” If the samples are listed, the SD card has mounted successfully.
   c. Exit the shell “exit.”
6. Install the malware scanning product to be tested.
7. Within the emulator window, find and launch the application.
8. Within the application, ensure that the most aggressive scanning option is enabled and that that the definitions are up to date. This will vary between applications.
9. Put the device in Airplane mode to prevent further updating.
10. Begin the scan.
11. Record the result. The detection rate is calculated based on how many of the 5560 samples are detected.

As it can be seen from Table 4, five of the malware scanners were unable to detect any malware samples. The authors decided to carry out further testing to investigate the possible cause by appending the .apk extension to a small set of the samples on the SD card. Removing the file extension is common
practice when sharing malware samples as it is thought to help prevent accidental installation of the malware. The samples selected were:

- 000a067df9235aee987cd1e6b7768bcc1053e640b267c5b1f0deeeef18be5d6be1.apk
- 000e094817bdec2b6e19d0f03e23f37910676a9b7e7709954614bac79269c36.apk
- 00c8de6b3109c32b658c30d7227488d2bce5353b31bedf5461419ff463072d.apk
- 00ceaa5f89be7a9ce5ffe96b5b6fb2e7e73ad87ecf023db9fa399c40ac59b62.apk
- 00cf11a8b905e891a4545e5b3fcae41f3ed405e3c5df9e1f3e310de4a88c42d0.apk

The five samples were chosen alphabetically when the malware set was listed. The results on this additional testing are shown in Table 4.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Detected Malware</th>
<th>Detection Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1CM</td>
<td>5532</td>
<td>99.49640287769784</td>
</tr>
<tr>
<td>2Q360</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3AVG</td>
<td>5546</td>
<td>99.74820143884892</td>
</tr>
<tr>
<td>4McAfee</td>
<td>5468</td>
<td>98.34532374100719</td>
</tr>
<tr>
<td>5AVAST</td>
<td>4846</td>
<td>87.15827338129496</td>
</tr>
<tr>
<td>6Norton</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7Lookout</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8NQ</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9Malwarebytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10Kaspersky</td>
<td>5420</td>
<td>97.48201438848921</td>
</tr>
</tbody>
</table>

Table 5

<table>
<thead>
<tr>
<th>Product</th>
<th>AV1</th>
<th>AV2</th>
<th>AV3</th>
<th>AV4</th>
<th>AV5</th>
<th>AV6</th>
<th>AV7</th>
<th>AV8</th>
<th>AV9</th>
<th>AV10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection rate</td>
<td>96.41</td>
<td>93.71</td>
<td>84.66</td>
<td>84.54</td>
<td>78.38</td>
<td>64.16</td>
<td>48.50</td>
<td>48.34</td>
<td>9.84</td>
<td>3.99</td>
</tr>
</tbody>
</table>

6. RESULTS AND ANALYSIS

The results obtained are illustrated in Table 4. Both AVG and CM scored over 99%; only failing to detect 16 and 28 pieces of malware respectively out of the 5560 samples. McAfee and Kaspersky had lower, but still high detection rates. Of the malware scanners that detected the malware, AVAST scored the worst, failing to detect over 1 in 10 (87%) of the malware samples. Q360, Norton, Lookout, NQ and Malwarebytes were unable to detect any of the malware samples.

The DREBIN paper tested ten antivirus scanners “(AntiVir, AVG, Bit-Defender, ClamAV, ESET, F-Secure, Kaspersky, McAfee, Panda, Sophos)” (Arp et al., 2014) against the same set of malware samples; their detection rate findings for these products are summarised in the table below. It is important to note that the VirusTotal (VirusTotal 2015) website was used to calculate detection rates in this case.

Direct comparisons to the DREBIN results cannot be made as VirusTotal does not differentiate between vendors’ offered
products. Malware scanning engines from AVG, Kaspersky, and McAfee were, however, tested in both experiments. The results are compared in Table 6 for completeness.

Table 6
Additional testing

<table>
<thead>
<tr>
<th>Product</th>
<th>AVG/3A VG</th>
<th>Kaspersky/10 Kaspersky</th>
<th>McAfee/4 McAfee</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREBIN Paper</td>
<td>93.71</td>
<td>48.50</td>
<td>48.34</td>
</tr>
<tr>
<td>Experiment Result</td>
<td>99.74820143</td>
<td>97.482014388489</td>
<td>98.34532374100719</td>
</tr>
<tr>
<td>Difference +/-</td>
<td>3884892</td>
<td>921</td>
<td>4100719</td>
</tr>
</tbody>
</table>

Further testing, using a set of malware samples appended with the .apk extension, gave the results presented in Table 7 when tested against the malware scanners that were unable to detect any of the malware samples (2Q360, 6Norton, 7Lookout, 8NQ and 9Malwarebytes).

Table 7
Testing using a set of malware samples appended with the .apk extension

<table>
<thead>
<tr>
<th>Reference</th>
<th>Detected Malware</th>
<th>Detection Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2Q360</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6Norton</td>
<td>4</td>
<td>80</td>
</tr>
<tr>
<td>7Lookout</td>
<td>5</td>
<td>100</td>
</tr>
<tr>
<td>8NQ</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9Malwarebytes</td>
<td>5</td>
<td>100</td>
</tr>
</tbody>
</table>

As Table 7 shows, two out of the five malware scanners (2Q360, 8NQ) showed no improvement in their detection rate when tested against the “.apk malware samples.” Three out of the five tested malware scanning products (6Norton, 7Lookout, 9Malwarebytes) began detecting the samples as malware once the apk extension had been added to the samples’ filename. This demonstrates that these products are not scanning the samples that do not have an apk file extension. This is a critical security risk as Android does not require a file extension to install an application. With some basic testing using Android virtual devices, the authors discovered that when a user opens an application file without an extension, the Android operating system prompts the user to choose an application to launch that file. The top (and default option) in this case is the Android package installer. It appears as though Android uses the file header rather than the extension to ascertain this. As a recommendation, the developers of these products should carry out the following to remedy the situation:

- Malware scanning products should scan all files by default and examine the file header to ascertain the file type. The filename and extension should not be relied upon.
- Malware scanning products should check these files against a set of known malware signatures; again the filename and extension should not be relied upon to do this as they are easily changed.

Previous major work (DREBIN) had relied on the use of VirusTotal, a third party analysis site, which the site owners themselves state is not fit for the purpose used. In regards to the static analysis we performed, in which the malware is not executed, the only factor affecting the detection rate would be the anti-malware application malfunctioning when run on a virtual device as opposed to a physical one. In all cases the applications did not report errors when installed, executed, or when performing scanning on the virtual device.

7. CONCLUSIONS AND FUTURE WORK

This paper presents the threats of malware in the Android environment. The results show
that half of the Android malware scanners were initially unable to detect any of the 5560 malware samples. Further testing proved that the reason for this in three of those products was that the file extension was used, in part, to regulate the scanning. This is a substantial security risk for two reasons: i) a file extension is easily changed and ii) Android does not require a file to have an .apk extension to install it as an application. This paper provides a snapshot of the current detection rates of the tested products and identifies a number of key flaws that could be abused by malware creators.

In terms of future work, it would be useful to investigate those products that showed no improvement in detection rate during the second part of the experiment. The authors would also like to further analyse the results of the top five malware scanners tested in this paper. It would be interesting to see if there are any commonalities amongst the samples that those scanners were unable to detect.

Finally, this paper focused on static analysis of the malware samples. Static analysis is the first line of defense against malware before it is installed or has the chance to be propagated by users. Many of the products tested claim that they will perform analysis of samples at runtime - detecting and blocking malicious behaviour. By testing both the static and dynamic capabilities of malware scanners, a more comprehensive detection rate could be calculated.
REFERENCES


