WM0824TU Economics of Cyber Security

Metrics Involved In The Security Assessment Of Mobile Application Distributors

GROUP 5

Raditya Arief	*4500318*
Eren Celik	*1550985*
Ana Guerra	*4520084*
Luigi Tuttobene	*4522729*

TU Delft

Since 2008, when Android Platform for mobile devices was launched due to the lliance between Google and Open Handset (Mahmood et al., 2012), android smartphones, and therefore android applications, have grown to the largest global market share. According to Google, 1.4 billion android devices are activated worldwide, and 2.4M apps are available on Google Play in September 2016. However, as Android smartphones become the most popular mobile device in the current market, Android security has received a lot of public attention in information security (Fang, Han, & Li, 2014). For instance, recent studies have shown that mobile markets are hosting malicious or vulnerable apps leading to compromise millions of devices, and therefore consumers.

In this paper, first, we introduce some security issues of Android applications and its implications in the safety of Android users. Second, actors involve in the software application landscape and their perceptions of ideal metrics are presented. In addition, a collection of the existing metrics in practice is shown. Then, we design metrics from our data set, which can help to make security decisions to stakeholders, after that, we evaluate these metrics and its usefulness. Finally, some conclusions are derived.

Security Issues Within Android Environment

Android apps have reached the large consumer market because small entrepreneurs and large software development companies were given the power and freedom to create enthralling mobile applications. In addition, those applications are easily delivered to the end-users through Mobile App markets. As a result of this software supply model, there has been an explosive growth in the number of new apps for Android. However, this accelerating growth also implies an increase in the security threats targeted at the platform, developers, and consumers (Mahmood et al., 2012).

Android applications are small programs that provide their functionalities by accessing and collecting sensitive data such as account passwords, contacts, financial records from users; accessing services located in the cloud such as Google, Facebook, or Twitter; and using device capabilities such as GPS, camera, microphone [(Bläsing et al., 2010), (Gilbert et al., 2011)]. Therefore, and due to the direct, and sometimes unlimited, access to user's sensitive data and Internet network, security issues of Android's applications can lead to undesirable outcomes for different stakeholders.

The security mechanism of Android, which can be considered as controls, has two levels. In the first level, Android apps execute in a separate Dalvik virtual machine, its own secure sandbox, as a result, apps are isolated from other apps in the memory. In the second level, Android restrict access to applications and critical resources using a permission based security model called Mandatory Access Control. The permission is a unique text string that is assigned to components and applications by Android or third party developers. Actually, more than 130 permissions are defined in Android operating system, ranging from access to camera (CAMERA), full access to the Internet (INTERNET), dialing a phone number (CALL_PHONE), and even disabling the phone function permanently (BRICK) [(Fang et al., 2014), (Mahmood et al., 2012)].
Although developers have to follow the security mechanism, it has been breached many times. For instance, when users install an infected or malicious app, the Android's standard security mechanism does not ensure protection to end users. The access permission in Android is coarse-grained (giving an application arbitrary accesses to certain

resources /high-level all or nothing permissions) thus, a malware embedded in an app can make use of all the access permissions that were granted to the host application (Mahmood et al., 2012).

According to Fang (2004), security issues in apps exist because of this Android's permission based mechanism. In his work, he identifies two categories of android security issues: direct and indirect. Direct issues, which include over-claim of permissions, permission escalation attack, and TOCTOU (Time of Check to Time of Use) attack, may lead to financial losses or leakage of user private information directly. However, indirect issues, including coarse granularity of permissions, incompetent permission administrators, and insufficient permission documentation, can be used as stepping stones in launching attacks to Android smartphones. Although developers and application markets should be aware of these security issues, users have to deal with its consequences.

Among the security issues that consumers, sometimes without being aware of, encourage, malware is the most dangerous. Three main social engineering-based techniques have being identified to install Android malware on mobile phones. First, repacking, which is one of the most common techniques used by malware authors to piggyback malicious payloads into popular apps. During the repacking, an App is downloaded, disassembled, enclosed with a malicious payload, re-assembled and submitted to official and/or alternative Android markets. Second, update attack, in this case, attackers do not enclose the payload as a whole, but instead only includes an update component that will download the malicious payload at runtime. Finally, drive-by download, which is nothing else than the traditional drive-by download attack but in the mobile world (Zhou & Jiang, 2012b).

Now the question is how malicious apps can threaten users? When malware is present in our devices, it can download and install new apps with or without user intervention. Monitoring and exfiltration of data, which means monitor and send away certain categories of data depending on the app's permission and control, it includes sensitive data as well as take control of the device capabilities. Activating premium services, making phone calls and sending SMS messages that incur charges for the user (Wei, Gomez, Neamtiu, & Faloutsos, 2012). Leaking content that can cause the passively disclose of private in-app data without any permission (Zhou & Jiang, 2012a). Moreover, Google's Android Security Review shows that malicious applications, such as ransomware, can restrict access to the owner's device until a sum of money is paid; and even that app can be used to perform denial of service attacks against other systems and resource (Google, 2015). In the latter case, the target is not the device's owner however, he or she is exploited to harm a third party.

Consequently, security issues of the Android apps have a huge impact on developers, market owners, consumers, end users, and even involve regulatory authorities.

Ideal metrics for decision makers

There are many different kinds of actors that are involved in the mobile app industries.

Few example of those actors is software developers, mobile app store owner, regulatory

authorities, and also the users themselves. These group of people is considered cyber security actors and they do make security decisions, even when they are not aware of it. Just like any other decision-making process, actors need data, or metrics, to support their judgment in making a right decision.
However, since every actor is involved in a different kind of process and pursues different objectives, they also require a different kind of data and metrics. It is also noteworthy that the ideal and the best security metrics are not always realistic or sometimes are very difficult to develop, which is why this issue is still one of the research priorities in the cyber security field.

Users as an actor probably are the largest in number, but perhaps also the least obvious.
Assuming that the primary security objective of most users is to avoid any losses from using a mobile app, then several security metrics can be developed to help users achieve this goal. One metric that can be very useful for users is a security flag. The security flag can be a feature of the mobile OS or the app store. The security flag works by informing the user if a certain app can potentially incur security issue, whether from malware infection or privacy leakage. By having this information, the user can make a sound decision whether to install the app or not. Other metrics that might be useful are app and developer rating system. An honest review that is not manipulated can be very useful for the user to make security decisions.

The software developer is also part of the mobile app industries.
In regard to security issues in the app marketplace, the probable primary objectives of this group of actors could be to prevent their apps from getting malware infections. Malware infection may result in reputation damage. Metrics that can be very useful for software developers are those that can be used to compare the security aspects between several app marketplaces so that they can make decisions about to which market they are going to publish their apps. The example of the metrics could be (1) the percentage of repackaged apps, (2) the percentage of malware-infected apps, (3) the number of attacks toward the market, and (4) the market vulnerability rating which measures the security level of each marketplace. These metrics may help software developers publish their apps more safely.

The most recognizable actors of this industry are the mobile app store owner, for instance, Google with its Google Play, Apple with its App Store, and also the 3rd party app stores such as Baidu App Store and Shouji 360 Tcpdn.
An example of cyber security problem for an app store could be malware infections on its hosted apps. Most of the times, the app store owner will not be the one who directly affected by the malware. However, they could potentially lose their competitiveness if the competing app stores manage to offer better protection for their hosted apps. To achieve this goal, one of the ideal metrics is a benefits measurement of a security investment. This metric should be able to include the amount of losses that is prevented from a particular security investment and calculate it against the cost of the investment. If such metric can be developed, then the market owner will have the capability to measure the result of their security investment and also to compare which security investment they should be taking. However, this metric is known to be very difficult to calculate.

Another type of actor involved in the industry is regulatory authorities.
The primary objective of this group of people could be to formulate policies that reduce, or even eliminate, the cyber security issues. An ideal metrics to help create good policies will be those which can measure the benefits of each policy. The measurement could be in number of prevented attacks or also could be in number of euros saved by a particular policy.

As said above, some of the security metrics mentioned before are very difficult to measure. Some of them are even impossible to measure. The cyber security experts are still pursuing the holy grail of cyber security metrics: to measure the benefit of a particular security investment.

Thus, one might wonder what the point in developing these ideal metrics if they cannot be measured?
There are two answers to this question. First, what could not be measured using today's knowledge might eventually be in the future. Second, assuming the metric is forever immeasurable, then having an ideal metric in mind will at least guide our effort and research to develop a metric that most closely embodies the ideal characteristics.

Existing metrics in practice

Due to its elusive and wicked nature, the efforts to introduce reliable indicators to measure security has resulted over the years in a variety of constructs. In fact, as the point of view on the issue changes from actor to actor, so does the problem perception and formulation, together therefore with the object of the measurement.
The goal of this section is to explore the most common metrics that are involved in the security assessment of digital distribution platforms for mobile apps, as well as to set them into a theoretical framework.

Rainer Böhme, in (Böhme, 2010) elaborates a secure investment model which identifies three variables, security costs, security level and security benefits, and formalizes their mutual relations. Security level appears as a mediating variable between the costs and the benefits of security. As we move from the expenses to the advantages, the framework distinguish between metrics based on controls, on vulnerabilities, on incidents and on (prevented) losses.

In the case of mobile application distributors, the control mechanism consists in subsequent verification steps from the moment the developer provides the software until the commercialization of the app. In addition, aggregated and anonymized data sent from user devices are used to monitor the general state of the market. Android Security Year 2015 Review (Google, 2015) shows the diagram and explain into details the code review and feedback loop system Google has established.

There is however a growing literature that has begun posing care to third party markets, distributors that repackage applications from Google Play and sell them.
(Zhou & Jiang, 2012b) and (Zhou & Jiang, 2012b) explain how during this operation malicious lines of code can be insert to turn the application into a malware, attaching also interesting data about the most exploited vulnerabilities and the most common malware in Android ecosystem, and arguing therefore

that the number of app repackaged might be a candidate indicator.

To do that, they rely on the infection rate, a metric largely discussed in (Truong et al., 2014).

This rate, definable as the percentage of infected apps over the population of a market, is pretty general, but represent the first attend to assess the level of security of a market. To define this construct, it is necessary to provide a criterion through which assess the level of infection of a software. In the literature, popular choices are a set of AV and malware detectors such as Avast, Avira, Baidu-International, Kaspersky, Qihoo-360, VitusTotal and similar. Truong et al., (2014). analyses also this aspect in his paper, concluding about the result shown in *Figure 1* that "a significant fraction of each dataset contains malware samples included only in that set, leading to the question whether there is any common agreement about what constitutes malware".

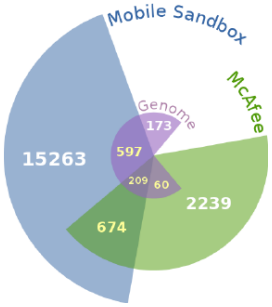

Figure 1: Sizes of the three malware datasets and the extent of overlaps among them. Source (Truong et al., 2014)

For these reasons, Ericka Chickowski recall the need of a second metric, defined in Dark Reading (2016) as False Positive Reporting Rate. Its aim would be to avoid the annoying foregoing situation. At closing look, more then a indicator of the security, this sound like a "meta-metric" to assess the reliability and the precision of the measuring tool.

Nonetheless, Greg Boison, director of cyber and homeland security at Lockheed Martin explains interviewed by Chickowski: "Despite the implementation of automated filtering, the [security operations center] team must make the final determination as to whether the events they are alerted to are real threats. The reporting of false positives to incident handlers and higher-level management increases their already heavy workload and, if excessive, can de-motivate and cause decreased vigilance".

Once agreed on a satisfying criterion of infection (that is likely to include a threshold to filter out the noise), the result will be something similar to what in (Kikuchi and al., 2016) is named Positive Detection Rate:

$$PDR(S,a) = |\{av \in S | av(a) = malicious\}| / |S|$$

where $av\ (a) = malicious$ means that an AV product av has detected application a as malicious, and S is the set of AV and detectors. At this point, it is possible to finally formalize the infection rate as malware detected ratio

$$MPR(m, S, M) = |\{a \in M | PDR(S, a) > m\}| / |M|$$

where M is the set of collected apps for a market and m is the chosen threshold value for the positive detection ratio.

Another interesting indicator discussed in (Kikuchi and al., 2016) is the Malware download ration, defined as

$$MDR(m,S,M) = \sum_M DL(a) / \sum_{M'} DL(a)$$

where $DL(a)$ indicates the total number of downloads of the app a. M' represent the set of all apps in the app set M which are determined as malware by the threshold m.

In light of this, the infection rate, MDR and similar metrics are half way between vulnerabilities and incidents, as they can be based on the vulnerabilities that might be

exploit or the number of application which are reported as malware (incidents).

A second class of metrics, based on the same constructs, aims to assess the security of mobile application distributors through the time span infected program remains online.
Ericka Chickowski (Dark Reading, 2016) distinguishes among average time to detection of a malware (ATD), average time to respond (ATTR) and mean time to fix software vulnerabilities.
Reducing ATD, personnel optimize the time to assess the situation, while ATTR refers to whether personal meets objectives to quickly and correctly respond to identified violations. In addition, Boison notes that decreasing ATTR will mitigate the impact, including cost, of security violations. As for the last one, developers should be measuring how long it takes to remediate software vulnerabilities from the time they are identified.
By keeping track of these three metrics continuously over time, these indicators provide a tool to assess progress in the in the detection and elimination of threats from the stores.

Designed Metrics For Dataset

The main tool used for the metrics and statistics are the VirusTotal libraries and the data set provided. The dataset consists of a list of most popular android apps on the Baidu and Qihoo 360 android app marketplace. The dataset is subdivided into 4 parts by separating both markets and subdivide it between two groups: "Games" and "Software". Games consist of any software meant for entertainment. The software group consists of any other software (mainly utilities and productivity software). Every app is also assigned a category that it corresponds with. The dataset consists of a selection of around 300 most popular apps within each category, totaling in over 50,000 android apps.

Without the use of some kind of malware detection the dataset is insufficient for any malware evaluation. For this end VirusTotal is used (virustotal.com). It is an online service for malware detection. It can check links/websites and uploaded files for any malware. VirusTotal is powerful because it pools resources from more than 50 anti-virus services to detect any malicious code. What one firm is not able to detect some others might. Therefor the results from VirusTotal is not only used to check for the presence of malware but can also be a useful check on how well it is detected among all the sources. This way, software can not only be checked for malware but also checked on how "elusive" the malware is.

The elusiveness of each infected application can be expressed in a 'malware detection ratio' metric. This is the simple ratio of sources that were able to detect divided by total sources queried.

The most obvious metric that can be produced with the datasets in conjunction with VirusTotal is the "percentage of the most popular applications which is identified as malicious." Since the listed apps in the datasets are already a population of the most popular apps from their respective categories, the calculation of the metric is fairly straightforward.

Malware presence ratio will be more telling of the intensity of malware presence. The data does not show how much malware is present

with each user. A user can have any number of infected apps on his/hers phone. This metrics is rather more an indicator for the ratio of infected transactions across the market. It says something about how much malware is actually downloaded. A market could have 50% infection rate but still have a no serious cyber security problem if downloads of infected apps is zero. Then malware does not have a notable role in the app market.

Metrics:
Malware detection ratio
1. The results from VirusTotal are used to determine if any malware is detected in an app.
2. Each app gets a detection ratio based on how many of the total sources recognize the malicious programming.
Malware presence ratio (% of all collected apps that are detected as malware)
The results from VirusTotal are used to view how many apps contain malware.
This number is then divided by the total amount of apps present in the market.

Malware download ratio
1. The results from VirusTotal are used to determine if any malware is detected in an app.
2. The download count of all infected apps are summed up.
3. The download count in the entire market is summed up.
4. Infected download count then is divided by the sum of the total market downloads.

The only way to scan such a large amount of applications through a site is by automation. VirusTotal provides a web API to make scripted processing possible. For bulk processing a private API key is necessary. Unfortunately due to circumstances it was not possible to receive a private API key in a timely fashion. Without a private key query rates were limited to 4 scans per minute.
This would mean more than 8 days for a single machine to go through the entire list. Another limitation to the public API key is that there is a file size limit of 32MB, which is an issue since a large part of the apps are >32MB. Because of these obstacles it was decided to only take a representative sample using a confidence level of 95% and a margin of error of 5%.
Taking standard deviation 0.5 (normal data) a representative sample size $N = 385$ is calculated. The sampling strategy is weighted sampling of each category. To get results applicable for the chosen metrics, care has been taken so that less downloaded apps are just as well represented as popular apps. This is done by sectioning each category into clusters of download counts. Within these clusters random samples have been taken with the help of excel.
The selected samples are then finally scanned for any malware and processed.

Defined Metrics In Practice: Evaluation

i) Malware Detection Ratio

Malware Detection Ratio is the average of malware infections of every app in a particular market. A single app can contain more than a single malware. This metric can be useful to the stakeholder of a marketplace. For example, the stakeholder of a certain market can determine the level of security control of its marketplace from looking at this ratio. The higher the Malware Detection Ratio, the worse the security control. However, this ratio is less

useful for the other security decision makers. The users might not be able to gather useful conclusion from this ratio. For example, a single malware infection might be no different from three infections because the users' objective is to avoid every malware infection, not to avoid as many malware as possible.

ii) Malware Presence Ratio

The Malware Detection Ratio is the rate of malware-infected apps in particular marketplace: the number of apps with infection divided by the total number of apps in its market. The ratio for each market can be seen in *Figure 2*.

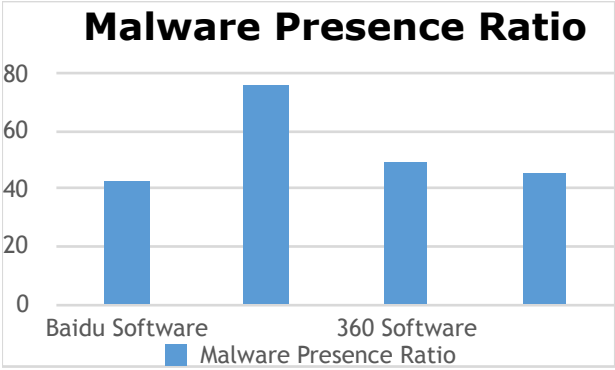

Figure 2 Malware Presence Ratio

This metrics is useful for the stakeholders of each marketplace. For example, the security decision maker from Baidu can tell that more than half of their hosted apps are infected by at least a single malware. This metrics is very useful to measure the effectiveness of an app marketplace malware control effort. Besides being helpful to the stakeholders of each marketplace, this metrics can also be useful to the users. By looking at the Malware Detection Ratio of each marketplace, the users can tell which marketplace is relatively safer. This metrics may also be useful to the regulatory authorities. For example, to protect the people from malware harm, the regulatory authorities can establish rules for maximum malware infection allowed. If particular marketplace's Malware Detection Ratio is over a certain limit, then the marketplace can be banned from operating. Overall, this metric is easy to digest and can useful for various security decision makers.

iii) Malware Download Ratio

The Malware Download Ratio calculate the ratio of malicious download divided by the total download of all apps for each particular app marketplace. The usage of this metric for the Android marketplace case can be seen in *Figure 3*.

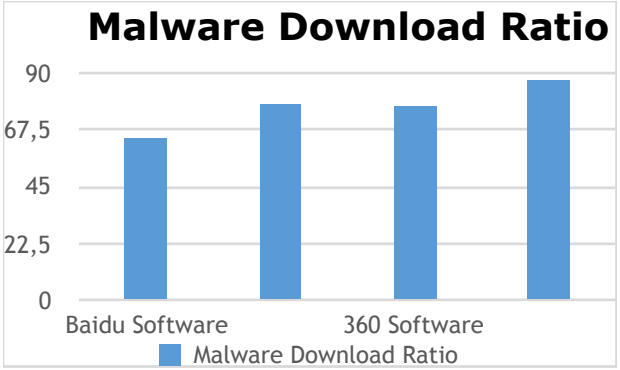

Figure 3 Malware Download Ratio

Although this metric can closely measure the extent of harm spread from a particular app marketplace, it is less useful than the Malware Detection Ratio. For various security decision makers, this metric can easily be replaced with Malware Detection Ratio. For example, it is less useful for users to know which app marketplace has the most "malicious" download; it is rather more helpful to them to know the ratio of infected apps instead.

	Baidu Software	Baidu Games	360 Software	360 Games
Total apps sampled	94	84	104	99
Clean apps	53	20	53	54
Apps with malicious content	41	64	51	45
Malware Presence Ration	0.436	0.762	0.49	0.455
Malicious app downloads	5.40E+09	1.50E+08	3.38E+09	5.48E+07
Clean downloads	3.03E+09	4.33E+07	9.95E+08	7.98E+06
Total downloads	8.42E+09	1.93E+08	4.37E+09	6.28E+07
Malware Download Ratio (MDR)	0.641	0.776	0.772	0.873
Average Detection % (clean apps counted)	6.93	20.49	5.72	9.92
Average Detection %	14.03	25.25	12.22	20.08

Table 1 Summary of results

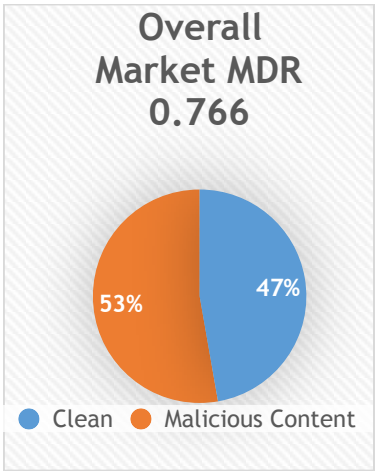

Table 2 Overall Market infection, MDR = Malware Download Ratio

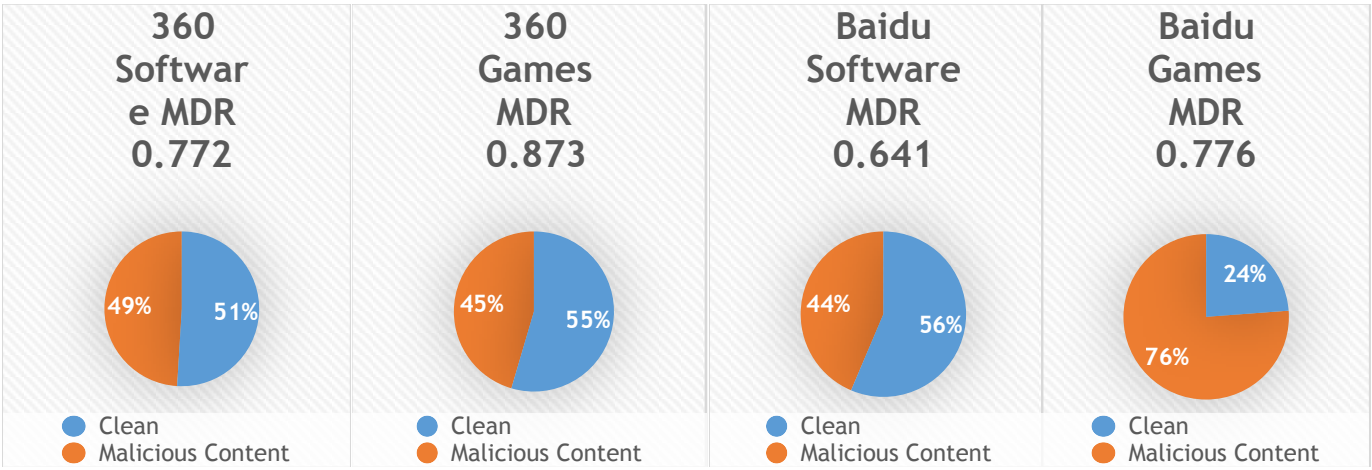

Table 3 Infection rate for the for segmented groups

Defined Metrics in Practice: Android Market Evaluation

Due to the functionality that Android application provide to users, apps have a direct access to the device's capabilities, user's data and information, and large networks such as Internet or communication service providers. Therefore, security issues and vulnerabilities in applications can be exploited for threat agents to damage different actors involve in this domain.

The results clearly show that the market is fairly compromised by various malicious programming as seen in table 1. On average there are more apps containing malicious code than clean apps. To make it worse the malware download ratio is always higher than the malware presence ratio. This means that malware is a bigger part of the eco system than that the malware presence ratio implies. Individually the Baidu Games group is the most compromised with 76% of the apps containing malicious code as can be seen in the pie charts.

The overall detection ratio lies around just short of 18% meaning that many sources VirusTotal queries are not able to detect the malware. This means that there is a good chance that virus scanners installed by users will not notice the malware which in turn worsens the cyber security problem.

It is fair to say that customers in these app markets have a high risk of being infected with malware of one kind or another.

Unfortunately it was not possible to scan all apps and without size restriction. The method used in this report might skew the results in an unpredictable way. On the other hand the dataset was already a pre-selection. It consisted of 300 most popular downloaded apps of each category. This in on itself is also a point to be taken in to consideration. The only thing that was not analyzed are the kinds of threats and their distribution. Gathering user data forms different threats and risks than infection with a destructive virus or ransomware. It is therefore advisable to do a complete scan and analysis of the dataset to give a more definitive answer to the question of how badly a market is compromised. Despite of this the results still point towards a massive infection of the market and consequently high probability that users are infected with one or more apps containing malicious code.

Bibliography

Bläsing, T., Batyuk, L., Schmidt, A. D., Camtepe, S. A., & Albayrak, S. (2010). An Android Application Sandbox System for Suspicious Software Detection. *Proceedings of the 5th IEEE International Conference on Malicious and Unwanted Software, Malware2010*, 55–62. http://doi.org/10.1109/MALWARE.2010.5665792

Böhme, R. (2010). Security Metrics and Security Investment Models. Advances in Information and Computer Security, pp.10-24.

Dark Reading. (2016). *Ericka Chickowski - Authors & Columnists - Dark Reading*. [online] Available at: http://www.darkreading.com/author-bio.asp?author_id=962 [Accessed 3 Oct. 2016].

Fang, Z., Han, W., & Li, Y. (2014). Permission based Android security: Issues and countermeasures. *Computers & Security*, *43*, 205–218. http://doi.org/10.1016/j.cose.2014.02.007

Gilbert, P., Chun, B.-G., Cox, L. P., & Jung, J. (2011). Vision: Automated Security Validation of Mobile Apps at App Markets. *Proceedings of the Second International Workshop on Mobile Cloud Computing and Services - MCS '11*, 21. http://doi.org/10.1145/1999732.1999740

Kikuchi, Y., Mori, H., Nakano, H., Yoshioka, K., Matsumoto, T., & Van Eeten, M. (2016). Evaluating Malware Mitigation by Android Market Operators. *Proceedings of the 9th USENIX Conference on Cyber Security Experimentation and Test*, 8.

Mahmood, R., Esfahani, N., Kacem, T., Mirzaei, N., Malek, S., & Stavrou, A. (2012). A Whitebox Approach for Automated Security Testing of Android Applications on the Cloud. *Ast*, 22–28. http://doi.org/10.1109/IWAST.2012.6228986

Truong, H., Lagerspetz, E., Nurmi, P., Oliner, A., Tarkoma, S., Asokan, N. and Bhattacharya, S. (2014). The company you keep. *Proceedings of the 23rd international conference on World wide web - WWW '14*.

Wei, X., Gomez, L., Neamtiu, I., & Faloutsos, M. (2012). Malicious Android applications in the enterprise: What do they do and how do we fix it? *Proceedings - 2012 IEEE 28th International Conference on Data Engineering Workshops, ICDEW 2012*, 251–254. http://doi.org/10.1109/ICDEW.2012.81

Zhou, Y., & Jiang, X. (2012a). Detecting Passive Content Leaks and Pollution in Android Applications. *NDSS Symposium 2013*, (October), 16. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.687.1281&rep=rep1&type=pdf

Zhou, Y., & Jiang, X. (2012b). Dissecting Android malware: Characterization and evolution. *Proceedings - IEEE Symposium on Security and Privacy*, (4), 95–109. http://doi.org/10.1109/SP.2012.16