Quantifying the Danger of Mobile Banking Applications on the Android Platform

Brett Ferris, Jay Stahle, and Ibrahim Baggili (Ph.D.)
UNH Cyber Forensics Research and Education Group
University of New Haven
West Haven, CT, USA
{bferr2, jstahle, ibaggili}@newhaven.edu

***Abstract*—The percentage of consumers that utilize mobile banking has increased in the past year and continues to grow. Mobile users are concerned about, but often do not understand the security risks that might be involved when conducting financial transactions through a mobile application. This research investigates application permissions and whether they are categorized as dangerous or normal as per the Android developer guidelines. In the absence of an existing risk index, we created ADI (Application Danger Index) based on these two types of permissions to help quantify the possible danger of each mobile banking application. Additionally a comparison was made between the percentage frequency of common permissions that occur in benign, malicious and banking applications to further inspect the potential danger in banking applications. Results showed that while banking applications did have an increased potential for danger, they more closely resembled benign applications than malicious applications.**

Keywords—Mobile Banking; Android; Risk

I. INTRODUCTION

In June 2007, Apple introduced the very first iPhone to the world. At that time they said the iPhone would revolutionize the term "Smartphone" [1]. Within one year of its debut, the iPhone (running iOS) introduced the public to the App Store, a platform that was user friendly that houses third party applications. Before the iPhone, a Smartphone was simply a mobile phone with basic Internet capabilities. Apple's new product integrated phone, music, and Internet communications into a single device while at the same time providing a platform for developers to create and distribute applications seamlessly to end users. It is this platform that shaped the nature of smartphones ever since.

In 2014, we take for granted the capabilities of our Smartphones. No longer are users simply looking for basic web surfing and email on their mobile device. Users want advanced mobile applications that enable them to be connected and conduct business wherever and whenever they travel. This demand is not limited to the Apple iPhone. According to Mahapatra [2], the Android Operating System (OS) has the largest market share of any mobile device OS in the world. The Google application marketplace, Google Play, has over 1,113,000 applications as of February 2014 [3].

A growing segment within the mobile applications market is mobile banking applications. Mobile banking increased by 33% from 2012 to 2013 and is predicted to see continued growth with increased smartphone use [4], [16]. A Federal Reserve survey found that concerns over security were a contributing factor that limited mobile banking adoption by consumers [4]. Users do not know what security concerns exist, and are wary of putting their financial resources in danger. Even though users are worried about the dangers of mobile banking, usage is increasing and banks are providing solutions to the demand for more applications and more features within those applications. It is a strange dichotomy that presents us with an interesting problem, and research questions. How can users understand the dangers of using mobile banking applications, mitigate that danger, and safely utilize the features they need?

One aspect of mobile application security is tied directly to the permissions built into the application itself. Both iOS and Android OS use a permissions based model for each application. This means that an application needs to ask the user's permission before it is able to execute a specific action. In the iOS system, one can download an application and then decide whether one wants to allow the use of certain permissions as they are being used by the application. For Android, one has to agree to allow access to all the permissions that the application is requesting before downloading. Additionally, with Android, one does not get to turn off any permission once the application is downloaded. The opposite is true in the iOS system [5]. The permissions required by an application can help present a view of the danger inherent in using that application. In this research, we focus on Android mobile banking application permissions. Our work explores what permissions are required, how dangerous those permissions are, and how mobile banking applications compare to known benign and malicious applications.

II. LITERATURE REVIEW

Since the increase in popularity of mobile applications there have been many studies on the security and permissions of applications. However, published work regarding application permissions seems to be based on the most popular applications regardless of their category. Felt, Porter,

Greenwood and David examined the permissions of 956 of the most popular Android applications and evaluated whether they are effective at protecting users. They found that 93% of free and 86% of paid applications had one or more dangerous permission/s. However, they also concluded that, when declared upfront by the developer, permission requirements can be beneficial to system security [6].

Chia, Yamamoto, and Asokan chose to base their studies on the most popular applications as well. They studied the link between user ratings and the privacy risks of applications as well as the number of permissions requested by Android applications. Their research showed that there is a small positive correlation between application popularity and the number of permissions requested. Applications that had a higher popularity requested a larger number of permissions. However, they go on to explain that this relationship is not strong enough to signal a privacy risk [7].

Other general studies have been conducted on mobile application permissions as a whole. Kelley, Consolvo, Cranor, Jung, Sadeh, and Wetherall [8] showed that users actually do not know the type of security risks that may be involved in allowing certain permissions for applications. Kelley et al. reported that users simply do not understand the permissions and what they entail, therefore they ignore them altogether.

While no published work focused on the security and permissions of banking applications, there has been one study conducted on mobile banking applications from a customer requirements viewpoint. Pousttchi and Schurig determined a defined set of customer requirements for mobile banking applications. These requirements come in four categories; Technical, Usability, Design and Security. They determined that the security requirement must include encrypted data transmission, authorization to data prior to usage, and an easy authorization process [9].

Another general study by Au, Zhou, Huang, Gill, and Lie looked at the permissions models across the major smartphone platforms. A comparison of permission models was performed between Android, iOS, Windows Phone 7, BlackberryOS, and Maemo. They assessed the permission models based on Control, Information, and Interactivity. The Android system was ranked medium in Control, high in Information, and low in Interactivity. iOS was ranked low in all three categories due to the lack of an explicit permissions system [5].

III. CONTRIBUTION

In our literature review, we failed to discover published work on the security of mobile banking applications. This paper contributes research that focuses on the security of mobile banking applications and their permissions specifically. We created an index to help users assess the dangers and security of their mobile banking applications. Our research provides a system for scoring mobile banking applications based on their requested permissions. The resulting index lets users know the possible danger level of each individual mobile banking application. This index also lays the groundwork for future indices that may reflect the probable dangers of banking applications more accurately.

IV. METHODOLOGY

The goal of this work was to learn what permissions are required for Android mobile banking applications, how dangerous those permissions are, and compare the applications to known benign and malicious applications. We executed five core steps required to accomplish these goals:

1. Identify applications
2. Identify permissions of each application
3. Identify android:protection level of each permission, i.e. Normal or Dangerous, and assign applicable value
4. Calculate the weighted total of danger and Application Danger Index (ADI) for each application
5. Calculate the rate of frequently requested permissions [10]

We used a Samsung Galaxy S4 running Android version 4.3 to download and inspect the test Android applications. This device was used to search for applications, download applications, identify permissions, and then identify the android:protection level of each permission. Additionally we used the Android applications RLPermissions [11] and Permissions [12] to assist in mapping the source code call for the permission to how it is displayed in the phone settings. For example, WRITE_EXTERNAL_STORAGE will be shown in the source code but will show up as *Modify or delete the contents of your USB storage* in the phone settings.

We started by identifying a list of the top banking institutions in North America. The website www.relbanks.com provided a list of the largest banks in America, listed in order by total assets [13]. This was used as a starting point to identify the mobile applications of each institution so that we could find the most common applications. The list of the largest international banks was also used to identify mobile banking applications from other countries that were available for download. We continued to add to our study by searching the Android marketplace for banking applications that may not have been on our initial list.

The next step was to analyze the actual permissions present in each individual application. When an application is installed on an Android device, the user is prompted with a list of all Normal and Dangerous protection level permissions required to run that application. However, only the Dangerous permissions are in view and the Normal permissions are available by selecting the "See all" option. This option allows one to view the Normal permissions as well. We installed each application and documented the listed permissions. This was a manual process for an important reason. We wanted to document the list of self-reported permissions for each application.

The Android marketplace terms of use require application developers to provide the list of required permissions in the

permission manifest section of the application [14]. This is the list that is then presented to the user. It was important for us to have the self-reported list of permissions. In the future, we could decompile the APK files of each application to find a definitive list of all permissions with API calls actually written in the source code. This would allow us to cross-reference the self-reported list with the extracted list for accuracy comparison.

The android:protection level of each individual permission characterizes the potential risk implied in the permission and indicates the procedure the system should follow when determining whether or not to grant the permission to an application requesting it [15]. The challenge presented by the android:protection level was in the difficulty of how to take a string rating system and convert it into a meaningful numerical value that can be analyzed. We formulated the following values:

- System/Signature - 0
- Signature - 0
- Normal - 1
- Dangerous - 5

We gave System/Signature and Signature values of (0) for two reasons. The first being that permissions with these android:protection level do not appear in the manifest or the self reported list of permissions. We have no way of identifying System/Signature or Signature protection level permissions without decompiling the APK file and analyzing the code directly. The second reason was that they are system level permissions that are not directly called by an application. That left the Normal and Dangerous android:protectionlevel designations to which we assigned a (1) and (5) respectively. Assigning these values allow us to calculate metrics for analysis. We were able to calculate a weighted danger for each application. This allowed a direct comparison of total danger between all applications. These weighted values also allow us to calculate what we call the Application Danger Index (ADI) for each application. This is an index that reflects danger per permission for each application. We calculated the ADI by adding the weighted value of each permission and dividing the weighted total by the number of permissions as shown in Formula 1.

$$\frac{Weighted\ Total}{Number\ of\ Permissions} = ADI \qquad (1)$$

The last step was to calculate the rate of frequently requested permissions [10]. This allowed us to directly compare our dataset of Android mobile banking applications to the known databases of benign and malicious applications published by Sarma [10]. This direct comparison allows us to analyze our pool of mobile banking applications against all Android applications to see if it more closely resembles benign or malicious software.

V. RESULTS

Our results formulate several ways to look at the security of a mobile banking application based on the requested permissions. Specifically, we take note of two categories of permissions: Normal and Dangerous. It is important to first understand what constitutes a normal and dangerous permission.

The website for Android Developers defines a normal permission to be "The default value. A lower-risk permission that gives requesting applications access to isolated application-level features, with minimal risk to other applications, the system, or the user. The system automatically grants this type of permissions to a requesting application at install, without asking for the user's explicit approval" [14]. It then defines a dangerous permission to be "A higher-risk permission that would give a requesting application access to private user data or control over the device that can negatively impact the user" [14]. Based on these categories we use our weighted scale to determine the overall danger and ADI. Comparing these two factors combined with the frequency percentage allows us to further examine the danger of an application.

A. Total Weighted Danger and Application Danger Index (ADI)

After each permission was given an individual weight, (1 for a normal permission and 5 for a dangerous permission) the total per application was calculated to provide us the Total Weighted Danger. This number represents an overall application danger. In most cases, this factor is directly proportional to the amount of permissions per application. For an application with more permissions, the Total Weighted Danger will usually be greater.

Fig. 1 shows the Total Weighted Danger from greatest to least for all (n=60) of the mobile banking applications that we examined. Out of the mobile banking applications that we examined, the one with the highest danger was the Citibank application. Citibank had a Total Weighted Danger of 71, which was 7 points higher than the next application, the USAA application. The lowest ranking Total Weighted danger was M&T Mobile application because it did not request any permissions. This application was simply a link to a mobile website where users can conduct their banking activities. The second lowest ranking application was Breeze (SCB), a bank from India, with a Total Weighted danger of 16, which was four points lower than the next highest ranked app.

Since the Total Weighted danger is highly dependent on the number of permissions, we formulated the Application Danger Index (ADI) to assist in determining the danger of the banking applications. Fig. 2 shows the ADI per application from highest to lowest. The State Street Bank Mobile, Zion Bank Mobile Banking, and Royal Bank applications scored the highest possible ADI with a 5 whereas the HSBC application scored the lowest ADI of 2.89 (aside from the M&T Mobile app which requested 0 permissions).

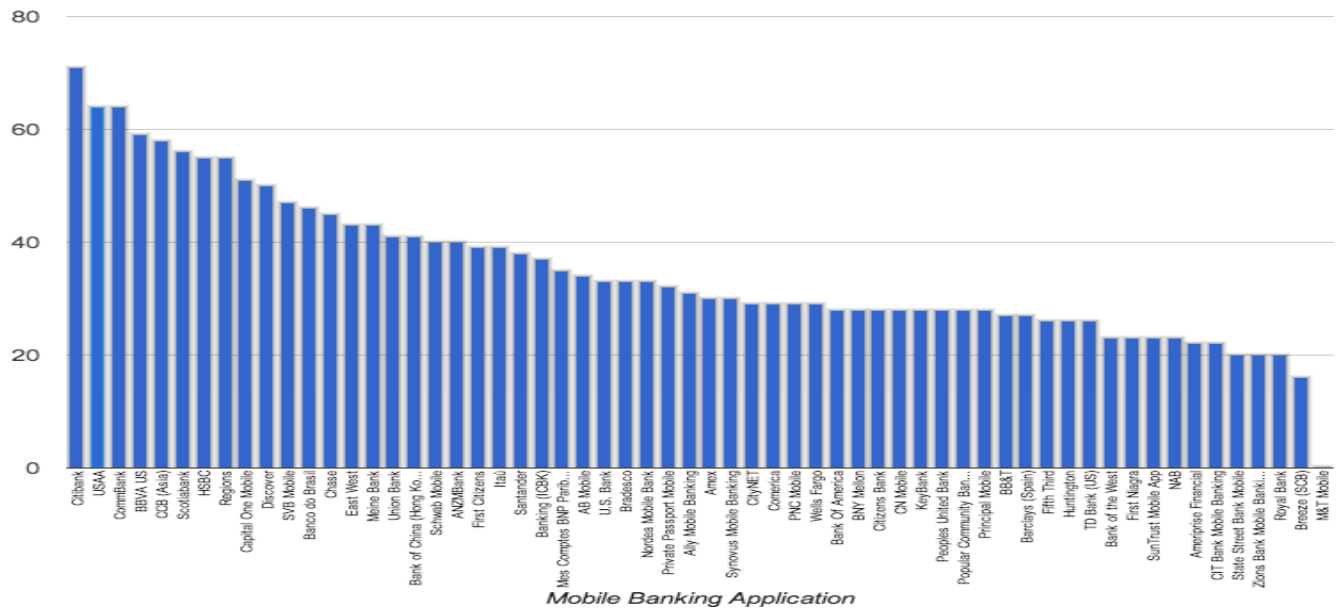

Fig 1. Total weighted danger per application

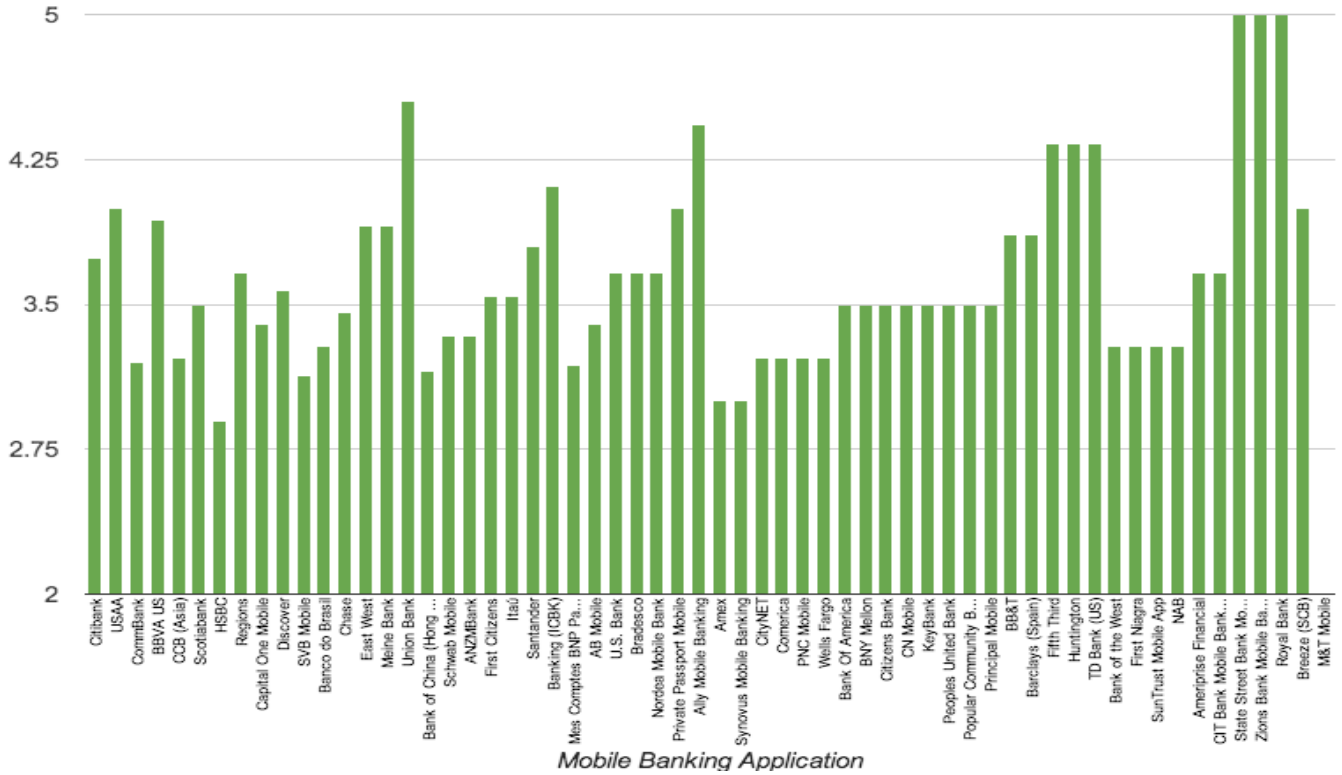

Fig 2. Application Danger Index (ADI) per application

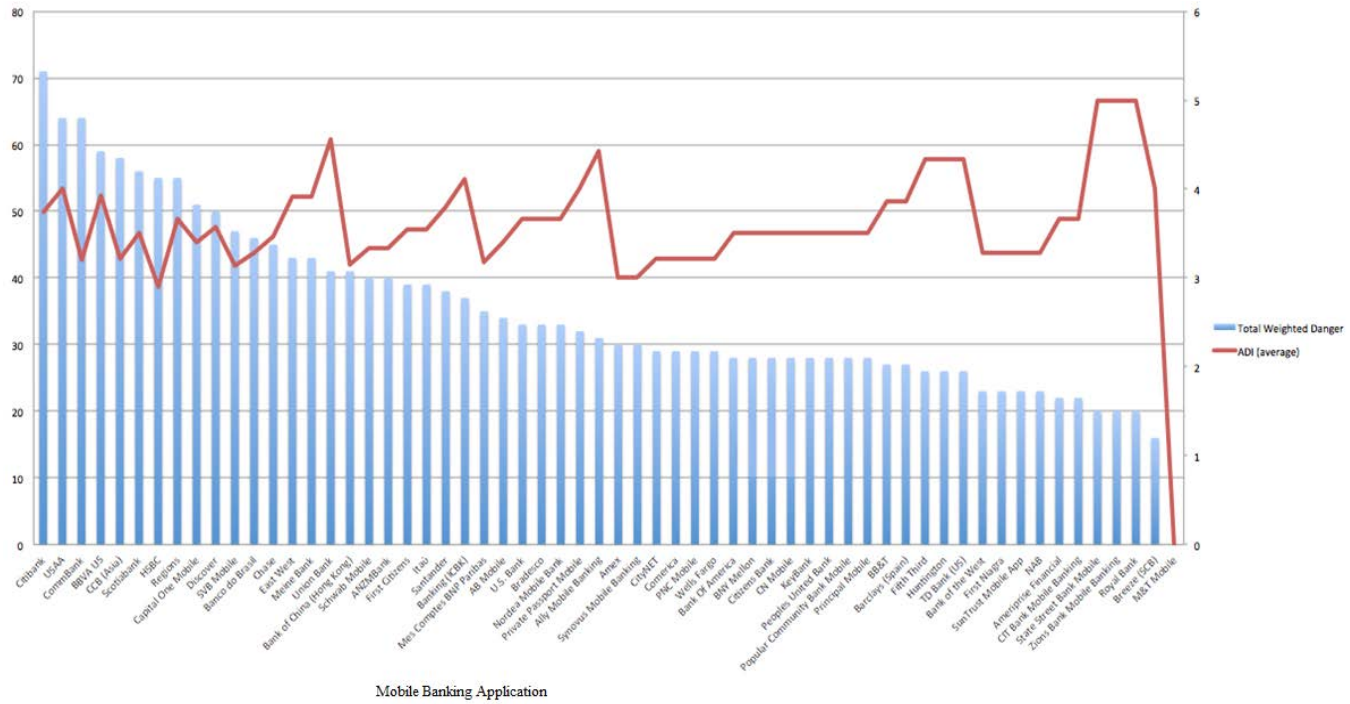

Fig 3. Total weighted danger and ADI overlayed

It is interesting to notice that while Citibank was the highest rated for Total Weighted Danger, it shows up with an ADI of 3.74 which is 19th overall. Similarly, the Breeze (SCB) application scored the lowest Total Weighted Danger, but scored a high ADI of 4.

B. *Percent Frequency*

Fig. 4 shows the combination of our banking dataset with the Sarma Table Permission Frequency [10]. We followed the same order of permissions in decreasing frequency of the benign dataset while matching the malicious and banking datasets. This allowed us to directly compare the frequency between all three datasets. Comparing our data with Sarma's, we see that there are no instances where the benign applications have a higher percentage than the malicious applications. We do see that the banking applications have a higher frequency than even the malicious applications in eight of the top twenty permissions. Additionally, we found an additional five out of the top twenty, for a total of thirteen out of twenty, where the banking applications exceeded the benign frequency but occurred less than the malicious.

A key finding in Fig. 4 is presented in the last five permissions listed. These five permissions are the five most common permissions found in the malicious dataset that are not present in the twenty most frequent benign permissions. This is important because it isolates and highlights the true differences between malicious and benign application. Of these five permissions, the benign applications appear three times and in very small frequencies, significantly less than the malicious. The banking applications dataset appears only once at significantly small frequency.

VI. CONCLUSION

This research discussed the security of mobile banking applications for the Android platform. Our research focused on e permissions requested by each of the examined banking applications. After manually extracting the permissions, we constructed our dataset consisting of (n=60) mobile banking applications. We assigned a value to each permission based on whether they were categorized as normal or dangerous and formulated an index we term the Application Danger Index (ADI) to help in determining the level of danger.

While the ADI is an assessment of security danger, it can be used in conjunction with the Total Weighted Danger and the Percentage Frequency of a permission to establish an even more in depth evaluation of the danger that might be present in a mobile banking application. For example we showed earlier that the Citibank application had the highest rated Total Weighted Danger. Looking at this value alone will tell us that the danger level is high. However, when combining that with the ADI for Citibank, which is 3.74, we determined that the danger level is not as severe as it first appeared. These two values tell us that this application must have a significant amount of permissions but a large percentage of them are normal or not dangerous because the ADI is low.

Furthermore, we were able to determine that our banking application dataset more closely resembled the benign application dataset than the malicious when compared to the five key permissions that identify the differences between benign and malicious.

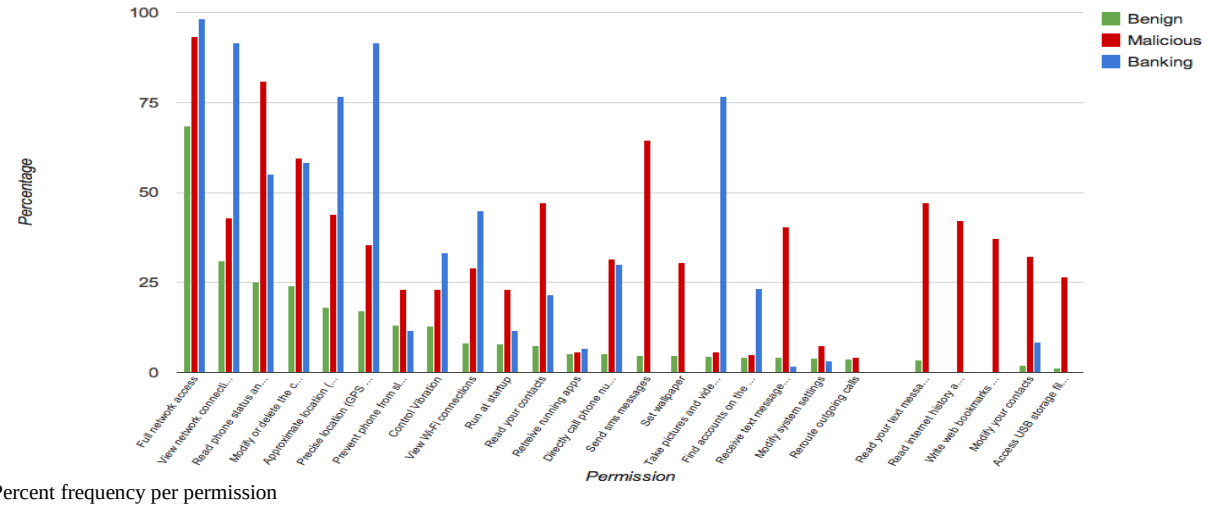

Fig 4. Percent frequency per permission

Our work quantified some danger and allowed us to compare banking applications to existing benign and malicious datasets, but it failed to succeed in creating a good index of risk. As they stand, we note that the Google Android security categories lack nuance and reliability. Future research should address this issue. A true Application Risk Index could be created by using a Delphi Method technique to analyze the risk factors of every Android permission. This would allow us to rank the risk of each permission on a Likert scale and create an accurate index for risk assessment. This was outside the scope of this initial research, but is essential to future research in the field. Mobile applications will continue to be adopted in greater numbers. Dependency on mobile applications for financial and personal transactions will also continue to increase. We note that a more accurate index for application risk assessment is needed.

REFERENCES

[1] [HD] Steve Jobs - iPhone Introduction in 2007 (Complete). (2013, January 10).YouTube. Retrieved March 1, 2014, from http://www.youtube.com/watch?v=9hUIxyE2Ns8http://www.youtube.com/watch?v=9hUIxyE2Ns8

[2] Mahapatra, L. (2013, November 11). Android Vs. iOS: What's The Most Popular Mobile Operating System In Your Country?. International Business Times. Retrieved March 1, 2014, from http://www.ibtimes.com/android-vs-ios-whats-most-popular-mobile-operating-system-your-country-1464892http://www.ibtimes.com/android-vs-ios-whats-most-popular-mobile-operating-system-your-country-1464892

[3] Number of Android Apps. (2014, February 13). Appbrain. Retrieved March 1, 2014, from http://www.appbrain.com/stats/number-of-android-appshttp://www.appbrain.com/stats/number-of-android-apps

[4] Gross, M., Rock, A., & Schmeiser, M. (2013, March 1). Consumers and Mobile Financial Services 2013.Federalreserve.org. Retrieved March 1, 2014, from http://www.federalreserve.gov/econresdata/consumers-and-mobile-financial-services-report-201303.pdf

[5] Au, K. W. Y., Zhou, Y. F., Huang, Z., Gill, P., & Lie, D. (2011, October). Short paper: a look at smartphone permission models. In Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices (pp. 63-68). ACM.http://www.bibme.org/http://www.bibme.org/

[6] Felt, A. P., Greenwood, K., & Wagner, D. (2011, June). The effectiveness of application permissions. In Proceedings of the 2nd USENIX conference on Web application development (pp. 7-7). USENIX Association

[7] Chia, P. H., Yamamoto, Y., & Asokan, N. (2012, April). Is this app safe?: a large scale study on application permissions and risk signals. In Proceedings of the 21st international conference on World Wide Web (pp. 311-320). ACM.

[8] Kelley, P. G., Consolvo, S., Cranor, L. F., Jung, J., Sadeh, N., & Wetherall, D. (2012). A conundrum of permissions: Installing applications on an android smartphone. In Financial Cryptography and Data Security (pp. 68-79). Springer Berlin Heidelberg.

[9] Pousttchi, K., & Schurig, M. (2004, January). Assessment of today's mobile banking applications from the view of customer requirements. In System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on (pp. 10-pp). IEEE.

[10] Sarma, B. P., Li, N., Gates, C., Potharaju, R., Nita-Rotaru, C., & Molloy, I. (2012, June). Android permissions: a perspective combining risks and benefits. In Proceedings of the 17th ACM symposium on Access Control Models and Technologies (pp. 13-22). ACM.

[11] Mehlmauer, C. (2013, November 19). Permissions. Retrieved April 1, 2014, from Google play: https://play.google.com/store/apps/details?id=com.FireFart.Permissions2

[12] RedLicense Labs. (2010, November 5). RL Permissions. Retrieved April 1, 2014, from Google play: https://play.google.com/store/ apps/details?id =com.redlicense.permissions

[13] World's Top Banks, (2013). World's Top Banks. Retrieved February 21, 2014, from http://www.relbanks.com/worlds-top-bankshttp://www.relbanks.com/worlds-top-bankshttp://www.relbanks.com/worlds-top-banks

[14] App Manifest. (2014). Android Developers. Retrieved March 1, 2014, from http://developer.android.com/guide/topics/manifest/manifest-intro.html

[15] Permission. (n.d.). Android Developers. Retrieved March 6, 2014, from http://developer.android.com/guide/topics/manifest/permission-element.htmlhttp://developer.android.com/guide/topics/manifest/manifest-intro.htmlhttp://developer.android.com/guide/topics/manifest/manifest-intro.html

[16] Fox, Susannah. "51% of U.S. Adults Bank Online." Pew Research Centers Internet American Life Project RSS. Pew Research Center, 7 Aug. 2013. Web. 16 Feb. 2014. <http://www.pewinternet.org/2013/08/07/51-of-u-s-adults-bank-online/>.