No Time At All: Opportunity Cost of Android Permissions (invited paper)

Gradeigh D. Clark, Swapnil Sarode, Janne Lindqvist
Rutgers University

ABSTRACT

App permissions detail the privacy-sensitive access to users' location, contact information, network access, and more. In this paper, we draw from the motivational work of McDonald and Cranor [18] on web privacy policies, to investigate the opportunity cost for users and the United States if people *would* actually read these permission screens on mobile devices during installation or launch time. We also demonstrate the time and cost differences between different versions of Android's permissions model. Based on our findings, an average user of Android M may spend less than two minutes annually viewing permission screens. This would mean a maximum annual opportunity cost of $1.29 and a minimum of $0.16 based on whether it was read at work or leisure. Reading detailed permissions screens in older versions of Android could require a user to spend nearly 90 minutes annually with a maximum cost of $67.24 and a minimum of $8.40. In our estimates, the United States would have to invest between a leisure cost of $36.67 million and a work cost of $293.6 million for Android M. However, these costs are small in comparison to costs needed in older versions of Android that could use up to as much as $1.87 billion in leisure costs $15.03 billion in work costs. We conclude that updates to Google's permissions layout has reduced the time and opportunity cost to the point where, with Android M, it is at an all-time low.

Keywords

Android, permissions, opportunity cost

1. INTRODUCTION

Mobile app stores aim to protect people's privacy by two main approaches: Apple by vetting applications before publishing to the App Store and Google by introducing a user-centric security model that requires developers to explicitly declare access to capabilities that Google has deemed sensitive. Android displays to the user the capabilities requested by an app during installation. The users' explicit task here is to decide whether to install the app or not.

This all-at-once model has lead to controversies with applications taking advantage of permissions to compromise user privacy. An example is when Facebook, Twitter, and other social media apps were deemed to be harvesting contacts [4] by copying entire contacts list to their internal databases. The Federal Trade Commission (FTC), a department of the United States federal government whose mission is to promote consumer protection, has looked into privacy concerns of mobile phone users as a result of these types of exploitative business practices. A takeaway of a 2013 report by the FTC was that operating systems should employ run-time disclosures at the instance a permission is being requested [9]. Google has since added run-time disclosures to certain privacy-sensitive permissions as of the release of Android M in October 2015.

Given how permissions can be exploited by companies, it becomes interesting to ask what investment would be required from users to read all the permissions being requested for the applications they download. In this paper, we investigate the opportunity cost for Android smartphone users *if they would actually read all the user permission screens*, with the goal of quantifying the opportunity cost burden placed upon users by Android as a result of their permissions models. As in previous work [17], we acknowledge that "few people read End-User License Agreements (EULAs) [12] or web privacy policies [14], because (a) there is an overriding desire to install the app or use the web site, (b) reading these policies is not part of the user's main task, (c) the complexity of reading these policies, and (d) a clear cost (i.e. time) with unclear benefit." [17] Thus, we evaluate what would the actual cost be in terms of time and money.

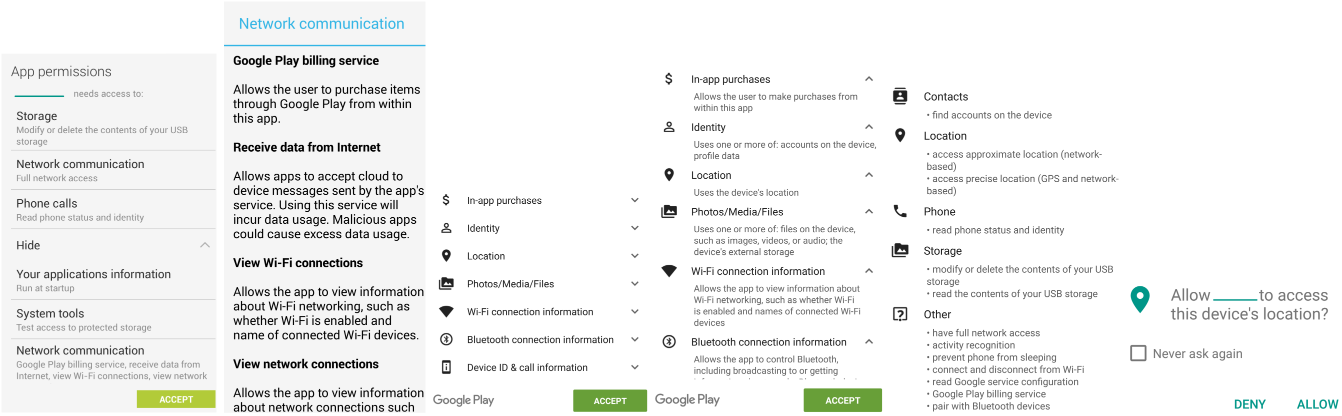
We build our analysis based upon previous work in the domain of opportunity costs from McDonald and Cranor [18], who demonstrated a similar analysis for the opportunity of reading web policies. We also extend this work to previous permission displays that Android has and quantify the changes in cost to users as a result of these updated permission models.

2. ANDROID PERMISSION SCREENS

The goal of the permissions are to show users potentially sensitive capabilities of the app. Android has gone through several iterations of its permission display model in the past few years in an effort to improve user awareness of what privacy-sensitive permissions are being requesting. Up until Play Store version 4.8.20, the information displayed on the

HotWireless'16, October 03-07, 2016, New York City, NY, USA

DOI: http://dx.doi.org/10.1145/2980115.2980117



(a) Explicit (b) Explicit(II) (c) Grouped (d) Grouped(II) (e) Grouped(III) (f) Request

Figure 1: The permissions displays under consideration. From left to right: explicit permissions model (Explicit) prior to Play Store 4.8.20, expanded explicit permissions (Explicit(II)) for ”Network Communication”, grouped permissions (Grouped) after Play Store 4.8.20, expanded grouped permissions (Grouped(II)) for all displayed categories, detailed group permissions (Grouped(III)) for the app on the Play Store, and a permission request (Request) for Location in Android M.

screen remained relatively static despite changes in UI elements (e.g. background color, font color) as shown in Figure 1a (Explicit). The permissions could be expanded to read specific descriptions of what was being requested, seen in Figure 1b (Explicit(II)).

Updates after 4.8.20 changed this to condense permission descriptions: detailed descriptions for the need of a permission were removed, and not all requested permissions by an application were shown at installation (Grouped Figure 1c), with expanded permissions in Figure 1d (Grouped(II)). Instead, Google moved the full list of permissions to a “Detailed permissions” link on the Play Store page for the app, shown in Figure 1e (Grouped(III)). For Android M, shown in Figure 1f (Request), users are no longer granting access to bundles of permissions at install-time but rather are queried about granting permission to specific privacy sensitive operations at the time the application needs to use them.

3. ESTIMATION METHODS

The opportunity cost of reading permissions is obtained as the time spent reading permissions multiplied by the estimated national average wage. This is shown by:

$$Cost = T_R \times w \quad (1)$$
$$T_R = n \times W \div R \quad (2)$$

where $Cost$ is the opportunity cost, T_R is the time it takes to read the permissions, w is the national average hourly wage, n is the average number of applications downloaded per year, W is the number of words on a permission screen, and R is the average national reading rate.

With these equations we can estimate the time taken to read the screens and the opportunity cost associated with that. To obtain the time taken for an entire nation we use the same, with the only difference being that the number of applications downloaded, n, is the number of apps downloaded by an entire nation in one year. We can use publicly available statistics to obtain values for w, n, and R. However, to get W, we needed to obtain values from a distribution of the word counts for commonly used applications.

We scraped the Play Store to accomplish this. For the current iteration of the permissions screen, Figure 1(c)-1(e), we scraped the top 100,000 applications on the Play Store and formed a word count for each of the different permission layouts. However, for the older versions displayed in Figure 1(a)-1(b), Google no longer offers a way to access them. We have access to a non-public dataset of permissions counts for the top ten applications in the twenty-six categories that was collected in 2013 prior to the permissions screen update. We use the data collected from these 260 apps for the older versions and for the newer versions we use data scraped from 100,000 applications in July 2016.

Different sources can provide different estimates for the variables listed in the equations. We create three different estimates for this – the *Lower Bound* represents the lowest possible value based on observation of data and sources, the *Upper Bound* represents the highest possible value, and the *Point Estimate* represents a midpoint between these two extremes. Our results are reported using the Point Estimate since it correlates best to the most common case.

4. FINDINGS

Phone Apps We begin by fixing a baseline for the number of applications downloaded by a user. Research2guidance is an analysis firm that reports the number of installs per month to be *two* [19], and Admob, a mobile advertising company, reports the value as *nine* [1]. We use these values as lower and upper bounds, respectively. We select five apps per month as a point estimate between the two boundaries.

Estimates	Monthly	Yearly
Lower Bound	2 apps	24 apps
Point Estimate	5 apps	60 apps
Upper Bound	9 apps	108 apps

Table 1: Permission screens visited in a year, estimated from publicly available data. Five apps is set as a point estimate between the lower and upper bound.

Word Count The length of the permissions screens vary depending on individual applications. To get an idea of the spread of different word lengths, we formed three quartiles at the 25%, 50%, and 75% levels. These quartile values correspond to splitting points of the distribution of word lengths. A 25% quartile value of 50 words means that 25% and less of the total words is less than or equal to 50. This is computed for every word configuration individually, as shown in Table 2, with corresponding labels matching with the labels in Figure 1. The three quartile levels respectively refer to the 25%, 50%, and 75% quartile levels in descending order.

Models	Lower Bound	Point Estimate	Upper Bound
Explicit	41	58	75
Explicit(II)	180	256	347
Grouped	4	8	13
Grouped(II)	25	55	89
Grouped(III)	49	88	128
Request	0	5	13

Table 2: Permission word counts displayed as quartiles (25%, 50%, 75%) of the distribution of total words. Explicit & Explicit(II) are computed using 260 apps from the older Play Store version whereas Grouped through Request are computed using a scraping of the top 100,000 apps in July 2016. Refer to Figure 1 for the permission models.

We take this information and combine it with Table 1 to compute annual word counts. When combining table information, we have a one-to-one mapping; Lower Bound multiplies into Lower Bound, Upper Bound into Upper Bound, and Point Estimate into Point Estimate. This gives us three numbers per permission model. The results of this is displayed in Table 3.

Models	Lower Bound	Point Estimate	Upper Bound
Explicit	984	3480	8100
Explicit(II)	4320	15360	37476
Grouped	96	480	1404
Grouped(II)	600	3300	9612
Grouped(III)	1176	5280	13824
Request	0	300	1404

Table 3: Number of words read annually for a given display model and quartiled word length. Each value is obtained by multiplying the estimated yearly apps from Table 1 by a corresponding quartile range in Table 2. An example: Grouped Upper Bound is obtained by multiplying Grouped Upper Bound by the Upper Bound yearly apps (108×13).

Reading Time The time to read a permission screen depends on two factors: the reading rate and the number of words. The reading rate for an average human is 180 words per minute (WPM) or three words per second (WPS) [5,21]. These average reading rates were computed without reference to the specific instance of smartphone permissions – which has icons for cues as well as variable font size on a smaller display. However, research on the influence of presentation on passages of text have not traditionally shown differences in reading rate [7,8]. Table 4 shows the amount of time it would take to annually read each permission screen, using this value of 180 WPM.

Models	Lower Bound	Point Estimate	Upper Bound
Explicit	5.5	19.3	45
Explicit(II)	24	86.8	208.2
Grouped	0.5	2.7	7.8
Grouped(II)	3.3	18.3	53.4
Grouped(III)	6.5	29.3	76.8
Request	0	1.7	7.8

Table 4: Number of minutes spent annually to read a given display model at a specific estimate. Each value is obtained by dividing the values in Table 3 by the average national reading rate, 180 WPM.

4.1 Opportunity Cost: Value of Time

The opportunity cost is the worth of the alternative option that could replace the activity under consideration. In this case, this represents the worth of other leisure activities or labor that could be done instead of reading the permissions required by an app. We estimate opportunity cost for leisure and work in terms of mean hourly wage. In the United States, overhead at work is twice the rate of take home pay [15]. This means that the overhead of reading the permission screens at work would cost twice the wage. Studies show that people estimate their leisure time at one quarter of their take home pay [16]. Based on this, we estimate the cost to read the permission screen at leisure as one quarter of the average wage. The United States Department of Labor reports the average hourly wage to be $23.23 [6]. Based on this, the estimated leisure cost is displayed in Table 5 and the estimated cost for work is displayed in Table 6.

Models	Lower Bound	Point Estimate	Upper Bound
Explicit	$0.52	$1.86	$4.34
Explicit(II)	$2.32	$8.40	$20.14
Grouped	$0.05	$0.25	$0.76
Grouped(II)	$0.32	$1.77	$5.16
Grouped(III)	$0.63	$2.83	$7.43
Request	0	$0.16	$0.75

Table 5: The total leisure cost for reading permissions annually for an average person as a function of the permission model. This is obtained by taking a quarter of the average wage [16], $23.23, and multiplying that by the reading times converted to hours in Table 4.

Cost to a Nation The opportunity cost to the United States can be calculated from the percentage market share of the nation out of the global share. In May 2016, at Google I/O, Google revealed that the download rate for mobile apps from the Play Store clocked in at 65 billion annually [20]. The United States contributes to 21% of the total downloads according to a report by research firm, App Annie [3]. Using this information, we can estimate that the United States accounts for 13.65 billion app downloads from the Play Store. The cost to the nation is determined by computing the Short, Medium, and Long word counts for 13.65 billion apps and converting that to wages as done in Table 5 and Table 6. The results of the leisure cost and work cost to the United States is shown in Table 7 and Table 8.

Models	Lower Bound	Point Estimate	Upper Bound
Explicit	$4.23	$14.96	$34.83
Explicit(II)	$18.58	$67.24	$161.21
Grouped	$0.41	$2.06	$6.03
Grouped(II)	$2.53	$14.19	$41.35
Grouped(III)	$5.05	$22.71	$59.46
Request	0	$1.29	$6.03

Table 6: The total work cost for reading permissions annually for an average person as a function of the permission model. This is obtained by taking doubling the average wage [6], $23.23, and multiplying that by the reading times converted to hours in Table 4.

Models	Lower Bound	Point Estimate	Upper Bound
Explicit	$300.9	$425.7	$550.5
Explicit(II)	$1321	$1879	$2547
Grouped	$29.4	$58.7	$95.4
Grouped(II)	$183.5	$403.7	$653.3
Grouped(III)	$359.7	$645.9	$939.5
Request	0	$36.67	$95.4

Table 7: The total leisure cost for reading permissions annually for the United States as a function of the permission model in millions of dollars. This is obtained by using the approximate number of apps downloaded by a nation, 13.65 billion, and computing the wage associated with the time spent reading permissions.

Models	Lower Bound	Point Estimate	Upper Bound
Explicit	$240.7	$340.5	$440.4
Explicit(II)	$10569	$15032	$20375
Grouped	$234.9	$469.8	$763.4
Grouped(II)	$1468	$3229	$5226
Grouped(III)	$2877	$5167	$7516
Request	0	$293.6	$763.4

Table 8: The total work cost for reading permissions annually for the United States as a function of the permission model in millions of dollars. This is obtained by using the approximate number of apps downloaded by a nation, 13.65 billion, and computing the wage associated with the time spent reading permissions.

5. DISCUSSION

We have estimated the cost of reading permissions for an individual annually as well as for the United States. The tables display values for the oldest version of Android's model (Explicit) to the most recent version (Request). Under the current iteration of the model, Request, the cost to the user has a point estimate of $0.16 for leisure (Table 5) and $1.29 for work (Table 6). Compared to the annual opportunity cost of reading web policies in 2008, $3,534 [18], this is a more manageable privacy burden. Even compared to the Upper Bound estimate of leisure cost, $0.75, and work cost, $6.03, the difference is still many orders of magnitude. It is far more costly and time consuming for a user to properly manage their privacy with web policies and EULAs than the Android permissions model.

We note that the lower bound cost of $0 for reading permissions in the current Android iteration, Request, represents a minimal cost scenario to users and even appears ideal at first glance. However, this is not is not feasible for a yearly value since the most popular applications in the store – and thus the ones most likely to be downloaded – usually access privacy-sensitive information in some way. The lower bound is capturing a collection of applications that still request permissions but do not trigger run-time disclosures. The Point Estimate or the Upper Bound for the Request category represents more likely cost scenarios.

When the price is scaled up to the United States, this represents a work cost of $293.6 million dollars and a leisure cost of $36.67 million under the current Android model, Request. Again, compared with web policies [18] having annual costs in 2008 to the United States of $781 billion dollars, these numbers represent a far more reasonable privacy request. The presumption here rests on the fact that users would read all of these permissions every time. There are also more potential, hidden costs associated with this that were not considered by us: the cost for mobile service, bandwidth costs, and storage costs on the device that would likely raise the overall amount considered in privacy disclosures.

A key observation about the differences in permission layouts is evidenced in Table 2. The original, verbose permission models had a much higher density of words compared to subsequent iterations. A direct comparison is available between the Explicit and Grouped models – the Explicit model displayed far more permissions on a screen than the more simplistic Grouped model. Even in the most detailed Grouped model, the detailed permissions inside the Play Store app shown in Figure 1e and referred to as Grouped(III) in Table 2, did not come close to the number of words in the most detailed Explicit model – shown in Figure 1b and referred to as Explicit(II) in Table 2. Google, as they iterated through Permissions models, was able to continually shave off word counts.

This creates an important take-away: that in a self regulated privacy disclosure model used in Android, Google has made significant headway in reducing overall costs as the permissions display has been updated. The choice of removing full permission descriptions and then grouping permissions in buckets at version 4.8.20 reduced reading overhead. However, the Grouped category does present challenges – a user looking for the most detailed information available would need to read through each display individually to guarantee full knowledge of what an application is doing. This would require, then, tallying the individual costs of each Grouped category to estimate the total cost of reading permissions to discover everything an application is requesting. The same is true of the Explicit model.

We acknowledge that users cannot be expected to read the permission screen of each and every application, since that is not the primary objective when they are installing and want to start using an app. Previous research [11] points to the direction that users are ignoring app permission screens and other warnings. Our work is asking the question *what if people actually would read the permissions, how much would it cost?* This is important because, for example, the US government is considering mandating different kinds of disclosures for mobile apps [9], raising the significance of what cost this represents to users.

Obvious further work would be to understand if the short form Request permission, in addition to being cost-effective, is effective in informing app consumers and helps them to make appropriate decisions. In our previous work, we have also explored alternative approaches to mobile app security & privacy, including explicitly showing users when their location is requested [11], showing network accesses of apps with Securacy [10], automated app analysis [2], crowdsourced privacy mental models [17] and privacy-preserving API design [13].

6. CONCLUSIONS

First and foremost, we are not claiming that all users would read all mobile app related permissions screens and privacy policies. Instead, we were asking what would be the cost if people would actually read all the screens given interest by regulators such as the FTC for mitigating privacy concerns [9]. Our contributions are in collecting information related to word counts for apps and using that to analyze various implemented permissions screens on the Android platform.

Our findings demonstrate that Google, through continual innovation of the permissions display model, has drastically reduced annual overhead time for individual users from a previous high of 86.8 minutes per user in their detailed Explicit model to 1.7 minutes for the latest version, Request, in Android M. In terms of cost represented to the United States, this a reduction from a high in the Explicit model of $1.879 billion in leisure costs and $15.03 billion in work costs down to $36.67 million in leisure costs and $293.6 million in work costs for Request model of Android M. We conclude that the cost of reading permissions screens is relatively small per user in the Request model of Android M and that Google has significantly reduced overhead cost over the years through redesign of its permissions display.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Numbers 1228777, 1223977 and 1546689. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Gradeigh D. Clark was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

7. REFERENCES

[1] Admob. Admob Mobile Metrics Report. https://techcrunch.com/2010/06/30/admob-android-apple/, 2010.

[2] S. Amini, J. Lin, J. Hong, J. Lindqvist, and J. Zhang. Mobile application evaluation using automation and crowdsourcing. In *Workshop on Privacy Enhancing Tools (PETools)*, July 2013.

[3] App Annie. APP Annie Index: Japan Overtakes U.S. For Google Play Revenues. http://blog.appannie.com/app-annie-index-november-2012/, 2012.

[4] BBC News. Social apps 'harvest smartphone contacts', Feb. 2012. http://www.bbc.co.uk/news/technology-17051910.

[5] R. Biedert, A. Dengel, G. Buscher, and A. Vartan. Reading and estimating gaze on smart phones. In *Proc. ETRA '12*, pages 385–388. ACM, 2012.

[6] Bureau of Labor Statistics. Occupational employment statistics. http://www.bls.gov/oes/current/oes_nat.htm, 2015.

[7] R. P. Carver. Effect of a" chunked" typography on reading rate and comprehension. *Journal of Applied Psychology*, 54(3):288, 1970.

[8] R. P. Carver. Is reading rate constant or flexible? *Reading Research Quarterly*, pages 190–215, 1983.

[9] Federal Trade Commission. Mobile privacy disclosures: Building trust through transparency. http://www.ftc.gov/os/2013/02/130201mobileprivacyreport.pdf, 2013.

[10] D. Ferreira, V. Kostakos, A. R. Beresford, J. Lindqvist, and A. K. Dey. Securacy: An empirical investigation of android applications' network usage, privacy and security. In *Proc. WiSec'15.*

[11] H. Fu and J. Lindqvist. General area or approximate location?: How people understand location permissions. In *Proc. WPES'14.*

[12] N. Good, R. Dhamija, J. Grossklags, D. Thaw, S. Aronowitz, D. Mulligan, and J. Konstan. Stopping spyware at the gate: a user study of privacy, notice and spyware. In *Proc. SOUPS'05.*

[13] S. Jain and J. Lindqvist. Should i protect you? understanding developers' behavior to privacy-preserving apis. In *Proc. USEC'14.*

[14] C. Jensen and C. Potts. Privacy policies as decision-making tools: an evaluation of online privacy notices. In *Proc. CHI'04.*

[15] R. E. Kmetovicz. *New product development: design and analysis.* Wiley-Interscience, 1992.

[16] T. Leunig. Time is money: a re-assessment of the passenger social savings from victorian british railways. *Journal of Economic History*, 66(3):635, 2006.

[17] J. Lin, S. Amini, J. I. Hong, N. Sadeh, J. Lindqvist, and J. Zhang. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In *Proc. UbiComp'12.*

[18] A. M. McDonald and L. F. Cranor. Cost of reading privacy policies. *ISJLP*, 4:543, 2008.

[19] Research2guidance. Android Market Insights, Volume 7, 2011.

[20] T. Verge. Android users have installed more than 65 billion apps from google play in the last year. http://www.theverge.com/2016/5/18/11673942/google-users-number-2016-android-auto-wear-tv-io, 2016.

[21] M. Ziefle. Effects of display resolution on visual performance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 40(4):554–568, 1998.