Securacy: An Empirical Investigation of Android Applications' Network Usage, Privacy and Security

Denzil Ferreira; Vassilis Kostakos
University of Oulu
Community Imaging Group
[denzil.ferreira; vassilis]@ee.oulu.fi

Alastair R. Beresford
University of Cambridge
Computer Laboratory
arb33@cam.ac.uk

Janne Lindqvist
Rutgers University
WINLAB / ECE
janne@winlab.rutgers.edu

Anind K. Dey
Carnegie Mellon University
HCII
anind@cs.cmu.edu

ABSTRACT

Smartphone users do not fully know what their apps do. For example, an applications' network usage and underlying security configuration is invisible to users. In this paper we introduce *Securacy*, a mobile app that explores users' privacy and security concerns with Android apps. Securacy takes a reactive, personalized approach, highlighting app permission settings that the user has previously stated are concerning, and provides feedback on the use of secure and insecure network communication for each app. We began our design of Securacy by conducting a literature review and in-depth interviews with 30 participants to understand their concerns. We used this knowledge to build Securacy and evaluated its use by another set of 218 anonymous participants who installed the application from the Google Play store. Our results show that access to address book information is by far the biggest privacy concern. Over half (56.4%) of the connections made by apps are insecure, and the destination of the majority of network traffic is North America, regardless of the location of the user. Our app provides unprecedented insight into Android applications' communications behavior globally, indicating that the majority of apps currently use insecure network connections.

Categories and Subject Descriptors

K.6.5: Management of computing and information systems: Security and Protection.

Keywords

Network; applications; privacy; context; experience sampling.

1. INTRODUCTION

Recent studies [21,49] of mobile malware reveal that the incentives for acquiring personal data are on the rise (*e.g.*, data for advertisers, in-app billing fraud) and consequently, user data is increasingly at risk. Users can be affected in several ways, including data theft or corruption, annoyance, device damage or location tracking [44]. Unfortunately, the automated screening process used by mobile app stores to detect malware are unreliable and can be circumvented [32].

In fact, the proliferation of malicious applications on mobile app stores has raised concerns regarding the potential harmful collection of personal data [10]. Once on the smartphone, malware may attempt to spread to other devices and/or enable remote data access. Many important features of smartphones require Internet connectivity, and therefore disabling network connectivity is not a practical solution to the problems generated by malware. In addition, global mobile Internet usage doubled between 2010 and 2011 and is estimated to increase 18-fold by 2016 [12]. As such, personal data may become increasingly vulnerable to capture by unwanted third parties.

WiSec'15 , June 22 - 26, 2015, New York, NY, USA
© 2015 ACM. ISBN 978-1-4503-3623-9/15/06…$15.00
DOI: http://dx.doi.org/10.1145/2766498.2766506

More critically, mobile apps often provide the user with explicit network usage actions such as loading a webpage or sending email, however, nearly 70% of network traffic generated by a device is invisible to the user [50]. On Android, users can check the volume of traffic sent but do not receive a breakdown of what and how data was sent and to where. In other words, it is challenging for user to recognize a suspicious application that may be generating unwanted and potentially harmful traffic.

To empirically investigate apps' covert network usage and better understand how users decide what permissions are acceptable for an application, we created a tool, called Securacy, that:

i) Provides insight into the applications' network connections that are established behind the scenes, including the use or otherwise of secure network connections and the geographical destination of the app data;

ii) Allows users to rate applications on potential permission abuse or use of insecure network connections, sharing the application's rating with all Securacy users;

iii) Notifies users of any potential network, security or privacy threat before first application use, based on shared ratings.

Our work has a similar motivation to Kepler [52], a browser extension that monitors and maps network data access, to raise users' awareness on widespread adware and spyware on desktop machines. However, on smartphones, mobile applications are provided with structured APIs that supply a significantly larger amount of sensitive user and sensor data when compared with browsers on desktop platforms.

When a user rates or uninstalls an application, we use experience sampling [13] to understand their concerns about mobile network usage, security and privacy. To the best of our knowledge, this is the first empirical study that investigates the combined effect of network security, server location and application permissions and user perceptions of applications' security and privacy.

We make the following contributions:

- A review on the exposed potential threats and systems that attempt to protect user privacy; and studies that investigate users' attitudes to privacy and security;

- A qualitative assessment of users' app installation decision process and users' security and privacy concerns;
- An intervention conducted in natural settings, deployed as an application that supports the users' mental model and mitigates users' security and privacy concerns;
- Insight into applications' network use and security: which servers do applications use the most and where are they located; and are they using secure or insecure network connections;
- An interpretation of the trust users place in applications and their rationale for uninstalling them; and the implications of our findings on future mobile applications' network use.

2. RELATED WORK

Smartphones come bundled with an increasing number of sensors. Besides mobile interaction (*e.g.*, using the accelerometer data to automatically rotate the screen), these sensors can be used to recognize its owner's context (*e.g.*, Google Now functionalities). However, privacy fears raised by the Snowden revelations and the fear of someone listening in [37,40] has given rise to additional concerns about smartphone sensors, for example, the "always listening" functionality on Moto X devices.

Android's security architecture uses a combination of permission- and signature-based checks. Consequently, an application needs to explicitly request access to some functionality (*i.e.*, application permissions), and an application can only be updated if digitally signed with the same certificate as the installed application. Unfortunately this approach has not prevented apps abusing the privacy of Android users. In general, iOS users are less exposed to privacy or security concerns, due in part to the mandatory code review process imposed by Apple on all submitted applications. However, malware does exist for iOS, but only 5% of iOS vs 20% of Android users are concerned about their privacy [7].

Mobile security and user privacy are challenging topics because of the diversity of potential threats. For example, AppIntent [54] revealed that applications can execute network operations without explicit user action. Fu et al. [26] found that applications increasingly fetch user location data without users understanding the reasons for it. Hill et al. [27] demonstrated how simple phone vibration and inaudible sound could be exploited to covertly share someone's device's information (*e.g.*, current location, phone number) with other nearby devices.

Sellwood and Crampton [47] discovered that permissions on newer APIs (including those exclusive to Google developers) would be automatically granted if the application was installed on a device prior to upgrading Android to a newer version. In other words, a malicious application could tentatively request future permissions (easily retrievable from Android's open-source code) and wait until the phone is upgrade to support them.

To protect user privacy and improve mobile security, researchers have proposed injecting fake sensor data [9,28], or device virtualization, albeit at the expense of performance [2]. Focusing on application permissions, Enck et al.'s TaintDroid [17] and VetDroid [56] both track the flow of personal data through an app in real-time. Enck et al.'s work revealed that the majority of the application network data is sent to advertisement servers without user consent or knowledge. This behavior is increasingly common, especially due to the increased availability of ad-supported games [4] and applications.

AppsPlayground [42] extends TaintDroid by monitoring suspicious internal system calls. RiskMon [31] modified the application initiation mechanism on Android to analyze the runtime behaviors of trusted applications, to discern them from potential malicious applications, while Apex [39] modified the permissions framework to allow users to select which permissions an application has access to, without aborting application installation.

Table 1. Summary of the various contributions in the last 5 years by related studies with applications for security and privacy.

Research goal		Year	Security	Privacy	Permissions	Modifications	Applications
Quantifying and classifying covert communications on Android	[28]	2014	•			•	
Continuous and automated risk assessment of mobile applications	[32]	2014		•	•	•	
Detecting and mitigating privacy leaks using crowdsourcing	[1]	2013		•		•	•
Enforcing user-defined security policies on untrusted applications	[4]	2013	•	•	•		•
Understanding of potential privacy leakages through apps	[5]	2013		•		•	
Automatic security analysis of smartphone applications	[42]	2013	•	•		•	
Exposing privacy-related behaviour	[46]	2013		•		•	
Analysing sensitive data transmission in Android for privacy leakage detection	[55]	2013		•		•	
Undesirable behaviors in apps with permission use analysis	[57]	2013		•	•	•	
Virtualization architecture for virtual smartphones	[2]	2011	•			•	
Mock'ing an application's access to a resource	[10]	2011		•	•	•	
Application behavior-based malware detection system	[11]	2011	•				•
Restricting permission access without loss of functionality	[29]	2011		•	•	•	
Information-flow tracking system for realtime privacy monitoring	[18]	2010		•		•	
User-defined runtime constraints	[39]	2010		•	•	•	

Kelley et al.'s [33] "privacy facts" prototype demonstrated that it is possible to improve user understanding of application permissions by increasing the clarity and display of the required permissions during the install dialog. AppProfiler [1] used anonymous feedback from users about application permissions to automatically generate a high-level summary (*e.g.*, acceptable, not acceptable, surprising).

AppGuard [3] is an Android application which recompiles Android applications installed on a phone handset. Recompilation inserts a security monitor that can be configured at runtime by the user to enforce privacy policies. Since AppGuard does not modify the operating system it does not require root or custom firmware to function. It does not work with all applications and the process needs to be repeated for every application update. It currently does not work on Android 5.0 and above (approximately 5.4% of the active Android devices [14]).

Table 1 contains a synthesis of the related work in the last five years that used applications to study *Security*, *Privacy* and Android *Permissions*; if the application requires *modification* to off-the-shelf smartphones; and if the application is *available* to the public. While effective at improving mobile security and protecting users' privacy, most of these approaches require OS modifications (13 versus 2, respectively), and hence these do not work on off-the-shelf smartphones. More critically, our literature review highlights that fact that the focus has been on exposing potential security and privacy threats, with little to no focus on user-friendly and easily deployable solutions. In our work, we aim at crowdsourcing anonymous feedback and data from users to create a security and privacy filter. In addition we do not require any system modifications: it is available as an application participants can download from the Google Play store.

3. Understanding users' application installation decision process and permission recall

To understand our requirements better, we conducted an information gathering study. We recruited our participants by affixing flyers around the University of Oulu campus. All participants were compensated with a three Euro coffee voucher. We conducted semi-structured interviews with 30 Android phone owners (9 female, 21 male), aged between 20 and 41 (M=28; SD=5) years old. The recruitment process targeted smartphone users rather than developers. All our participants were students, ranging from computer science to biology majors.

We created a questionnaire based on previous findings about users' privacy and security concerns [32,33,48] to capture the decision making process our participants went through when installing an application. We used a Likert-scale (1-totally disagree, 2-slightly disagree, 3-not sure, 4-slightly agree, 5-totally agree) for the following statements:

"When you install an application…"

- It is important to recognize who created the application;
- I trust my data to the developer I recognize;
- I always check the application's permissions;
- I worry about how safe my network connection is;
- I care about others' ratings on the application store;
- I rate applications I like.

To follow up, we asked participants to elicit from memory the permissions they check when installing an application. Lastly, we asked participants to select the data categories they would be comfortable sharing with any application. The data categories used are Google's recently updated permission groups[1] which follow Felt et al.'s [1] permission recommendations:

1. **Profile**: access to user profile (*e.g.*, name, social stream and subscribed feeds);
2. **Location**: access to GPS- or network-based location;
3. **Contacts**: access to contact list data;
4. **Documents**: access to documents (*e.g.*, pictures, music, storage, dictionary, logs);
5. **Calendar**: access to calendar events information;
6. **Messages and calls**: access to message and call information and history;
7. **Accounts**: access to device accounts (*e.g.*, Google, Facebook, Twitter);
8. **Browser**: access to browsing history and bookmarks;
9. **Network and Internet**: access to Internet, and networks' information (*e.g.*, Wi-Fi access points, connected Bluetooth devices).

We concluded with a semi-structured interview where we further clarified the rationale behind participants' answers, solicited further comment and asked if they had any questions.

3.1 Results and Analysis

For our statistical analysis, we grouped the ratings: 1 ("totally disagree") and 2 ("slightly disagree") as "Disagree"; 3 ("not sure") as "Undecided"; and ratings 4 ("slightly agree") and 5 ("totally agree") as "Agree." We do so to simplify analysis and ensure consistency between answers from different participants.

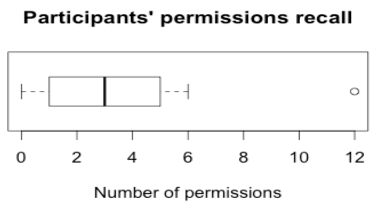
Similar to [8], we found that *recognizing* (Cockran's Q $\chi^2(2) = 13.4$, $p = 0.001$, $\phi = 0.66$) and *trusting* (Cockran's Q $\chi^2(2) = 18.2$, $p = 0.0001$, $\phi = 0.77$) the application developer are significant factors that are considered when installing an application. Contrary to Felt *et al.*'s [1] results, with a medium effect, our participants *do check* an application's permissions before installing it (Cockran's Q $\chi^2(2) = 9.8$, $p = 0.007$, $\phi = 0.57$). Similarly, participants worry about where application data is stored (Cockran's Q $\chi^2(2)=6.2$, $p = 0.04$, $\phi = 0.45$) and about how secure the network connection is (Cockran's Q $\chi^2(2)=7.8$, $p = 0.02$, $\phi = 0.50$). However, during our interviews, all participants confessed that they do not know where the data is actually stored, referring to "somewhere in the cloud," and do not fully understand how secure different networks are. One participant remarked *"…isn't the 3G mobile network secure? (P24)" i.e.*, thinking that using a data plan on its own is safe.

Paradoxically, participants largely rely on the Google Play ratings for making decisions (Cockran's Q $\chi^2(2)=43.4$, $p < 0.001$, $\phi = 1.20$) but do not rate applications themselves (Cockran's Q $\chi^2(2)=15.2$, $p = 0.0005$, $\phi = 0.71$).

Overall, our participants only recalled on average 3.3 (SD=2.63) permissions, with the most common being messages and calls (73.3%), contacts (46.6%) and location (43.3%). See **Figure 1**. When asked about how important the permissions were to them, 6 of the 30 participants said they "*[…]do not care about the permissions[…]*." Instead, they rely on feedback from others and

[1]https://support.google.com/googleplay/answer/6014972?p=app_permissions

ratings on the Google Play store. One participant responded that the most important permissions are "*those that cost money*," dismissing the rest.



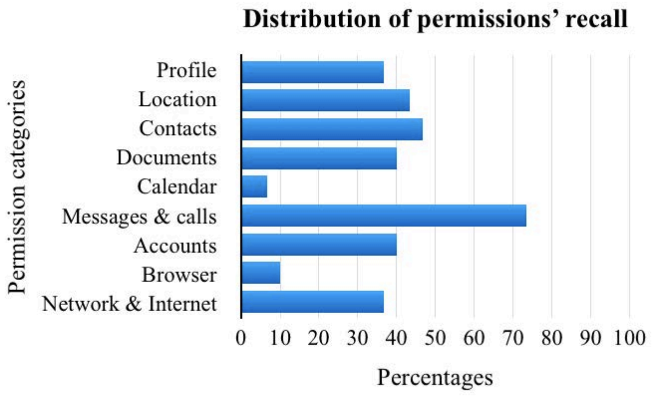


Figure 1. Participants' ability to elicit permissions from memory.

We found that the consolidated list of permissions affords a better recall of permissions and understandability of these permissions. Our participants are generally comfortable sharing some data and resources with applications, including Internet connectivity (83.3%), location (80.0%) and profile data (70.0%). On the other hand, they are least comfortable sharing access to their messages or calls (6.6%), their user accounts and contact list (23.3%) or their documents (30.0%) (Kruskal-Wallis $\chi^2(8)=76.5$, $p<0.001$). See **Figure 2**.

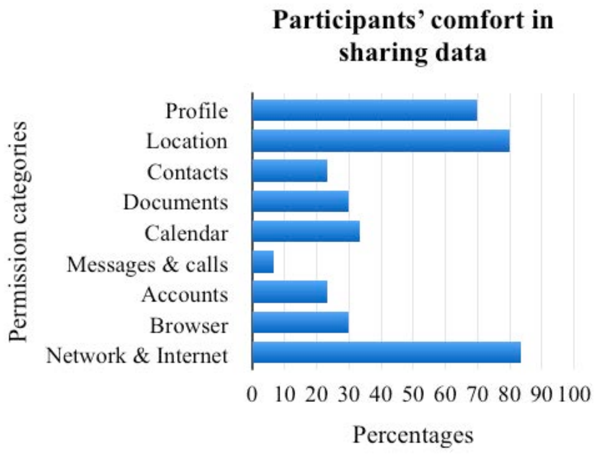


Figure 2. Participants' comfort in sharing data with applications per category.

In summary, we found from our questionnaire and semi-structured interviews that:

- Users do recognize and trust popular application developers;
- Users do not fully understand what some permissions do, why they are needed or for what are they used for;
- Users do not know where their data is being stored or where it is being sent;
- Users do not know whether network communications are secure;
- Users rely on others' feedback (*e.g.*, reviews, ratings) without necessarily rating themselves.

4. Securacy's Design, Implementation & Evaluation

We used our findings to design a security application that reduced the number of times users need to recall the underlying meaning of permissions. Instead our application only notifies users if an application they installed uses permissions they do not generally find acceptable. The tool also provides further insight into the location of the server an application communicates with, and whether the communication takes place over a secure network connection. In addition, it leverages users' reliance on others' feedback to evaluate an application. We further investigated the following research questions:

A. *Which servers do applications use the most and where are they located?*

B. *Are applications using secure or insecure network connections?*

C. *How do users rate an application from a security and privacy standpoint?*

To investigate these questions, we designed and implemented an app called *Securacy*[2] using AWARE[3]. AWARE is event-driven and does not affect device usage [22,23]. After deploying Securacy to the application store, we recruited 218 anonymous participants by advertising on Facebook and Twitter. We offered no additional compensation beyond use of the Securacy app itself. For a period of approximately 6 months (between January-June, 2014), we collected the following data:

- ***Foreground application***: the application the user is currently using;
- ***Installed, removed or updated application***: the applications the user installed, removed or updated;
- ***Application permissions***: the permissions an application requests from the application manifest;
- ***Application server connections***: the application-server connection information (*e.g.*, server IP address, the port used to connect and geo-IP location);
- ***User location***: captured only when a network connection is established;
- ***User permission concerns***: users' explicitly indicated concerns about data access;
- ***User ratings***: users' application ratings over time.

We explained Securacy's data collection process clearly on the Google Play storefront so potential users knew how we were processing their data. Our research protocol was reviewed and approved by University of Oulu's ethics committee. No personal

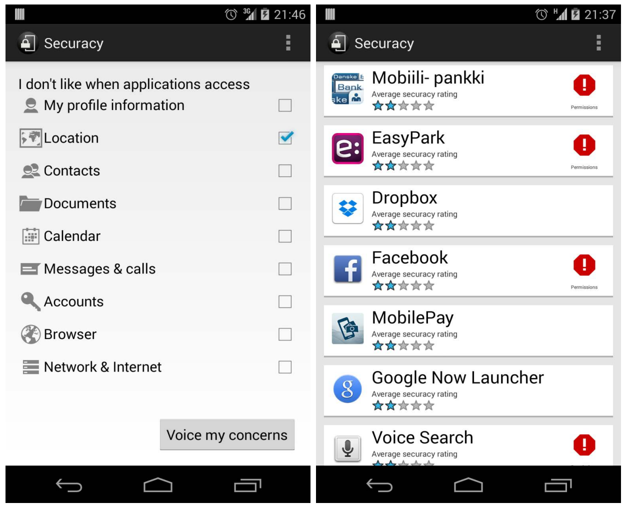
[2] Video: https://www.youtube.com/watch?v=sazpo_r8CZU

[3] Freely available at http://www.awareframework.com

identifiers (*e.g.*, phone number, emails) were collected at any time and the data was securely stored on our servers. Only the researchers involved in this work had access to the collected dataset.

4.1 Design & Implementation

The main interface of Securacy lists all installed apps, the public rating – an average of all users' ratings – for an app, and a red flag if the application requests one or more of the permissions a user is concerned about. A screenshot is shown in **Figure 3 - right**.



I don't like when applications access
- My profile information ☐
- Location ✓
- Contacts ☐
- Documents ☐
- Calendar ☐
- Messages & calls ☐
- Accounts ☐
- Browser ☐
- Network & Internet ☐

Voice my concerns

Mobiili- pankki Average securacy rating	Permissions
EasyPark Average securacy rating	Permissions
Dropbox Average securacy rating	
Facebook Average securacy rating	Permissions
MobilePay Average securacy rating	
Google Now Launcher Average securacy rating	
Voice Search Average securacy rating	

Figure 3. Expressing permission concerns (left) and the main interface (right).

Users' perspectives on what is secure and private are diverse, and it is thus challenging to assess automatically [6]. When users run our application for the first time, they have the opportunity to indicate what data they do not generally expect applications to access. A screenshot is shown in **Figure 3 - left**. These can be edited at any time from Securacy's main menu. For practicality and to reduce potential user burden in managing permissions for each application individually, we used the permission categories as an overarching permission filter for any application the user installed, removed or updated.

Requiring users to review a list of permissions immediately before installing every application has been reported as a nuisance, interrupting the installation flow [32] and is an administrative burden for the user [48,1]. In contrast to *e.g.* Lin et al.'s proposal that permissions' presentation on the app store should be modified to include rationale for their use [35], we focused on extending install-time functionality. Securacy acts unobtrusively as soon as an application is installed, and interacts only if needed with the user before the application is executed for the first time.

Concretely, when an application is installed or updated, Securacy checks its permissions automatically. Once a new application is installed, the user needs to explicitly launch an application for it to access any data on the device. Therefore, this just-in-time approach allows us to check the application's security and privacy standing according to the public ratings (*i.e.*, application's average rating from all Securacy users) before the user actually uses the application (Figure 4).

For reassurance, Securacy notifies the user if any of the permission concerns are triggered, and if available, displays the app public rating. If there is no rating, the user is encouraged to rate the app. At this point, when the user taps the notification, the user will be able to see preliminary information.

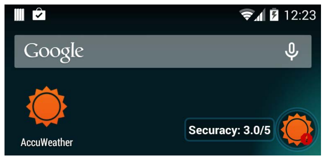
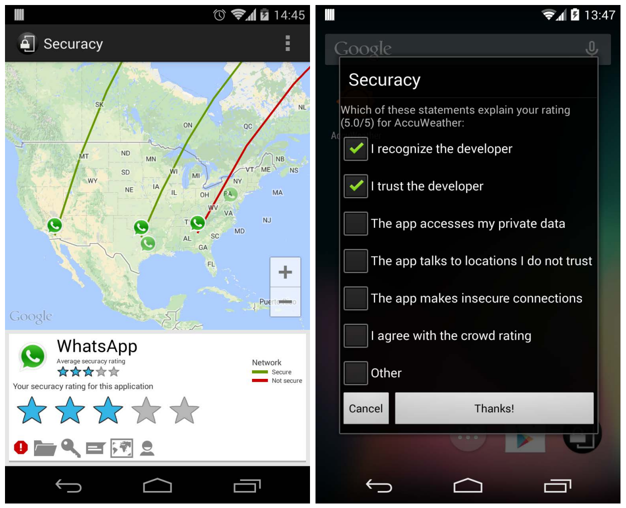


Figure 4. Example notification. AccuWeather is requesting a concerning permission, rated as 3 out of 5 by the public.



Securacy

Which of these statements explain your rating (5.0/5) for AccuWeather:

- ✓ I recognize the developer
- ✓ I trust the developer
- ☐ The app accesses my private data
- ☐ The app talks to locations I do not trust
- ☐ The app makes insecure connections
- ☐ I agree with the crowd rating
- ☐ Other

Cancel | Thanks!

Figure 5. WhatsApp security details (left). Example decision process questionnaire, shown after rating an application (right); note that selecting the "other" statement reveals a freeform text box to capture more detailed feedback.

Securacy reveals a range of information for each individual app, including the application name and icon, public rating, the user's own rating, any permission concerns, and the use of secure or insecure network connections. Securacy also shows a map of all known server connections for an application, collected from all Securacy users, regardless of whether it has been rated yet.

If the user rates an application for the first time, we capture the rating decision process using an in-situ mobile questionnaire. Example screenshots, including the questionnaire, are shown in **Figure 5 - right**. If the user uninstalls a previously rated app, we also use the mobile questionnaire to ask "Which of these statements explain why you removed [application name]"; we use the same explanations from the rating questionnaire as shown in **Figure 5 - right**. Open-ended data (*e.g.*, "Other") can also be collected by this questionnaire.

With AWARE, we capture when the user switches between applications. Each Android application has a unique installation ID (UID). To capture their individual network use, we monitor four system-generated, but world readable files – ***/proc/net/tcp(6)*** and ***/proc/net/udp(6)*** for IPv4 and (IPv6), respectively – and filter the matching application UID's to the network connection traces. From these connection traces, we extract the destination IP and connected port, and then classify it as secure over ports 22, 26, 443, 465, 563, 636, 695, 898, 706, 989, 990, 992, 993, 994, 995, 2087, 2096, 2484, 2949, 3269, 3424, 4843, 5223, 6619, 6679, 6697, 8443, 8883, 9091, 10302, 12975, and 32976; and insecure otherwise, per IANA specifications [30].

4.2 Evaluation "in-the-wild" of Securacy

During a 6-month long deployment, the 218 anonymous participants reviewed a total of 219 distinct applications and provided us with a total of 406 ratings. We also collected 26 questionnaires during application removal. We collected 41,794 unique network connections from 422 distinct applications.

A. Which servers do applications use the most and where are they located?

The majority of the servers used by applications were located in North America (65.5%), with 23.3% in Europe, 7.0% in Asia, 1.5% in South America, 1.4% in Africa, and 1.3% in Australia (**Figure 6 - left**). We found the largest number of connections (4,077 out of the 41,794) established to a Facebook server in North America; and 1,900 to Facebook in Europe. Applications with advertisements made 1,771 connections to a Google's AdMob server; and we recorded 1,544 connections to Amazon's elastic cloud hosting service. Our fifth most recorded connection was to our own server, which participants devices connected to on 1,430 occasions to synchronize Securacy data and submit ratings (**Figure 6 - right**).

During the field deployment, participants' applications established 115.0 (SD=282.5) network connections while using 10.9 (SD=12.7) applications on average per day. The majority of our participants are from Europe (41%) and North America (53%). A network analysis across all applications revealed that applications connect to North America the most, independently of the country of origin (**Table 2**).

Some port numbers are used to support both secure and insecure connections. For instance, STARTTLS enables applications communicating with some protocols (e.g., SMTP) to upgrade an insecure connection (e.g., port 25) to a secure one without reconnecting to a different port number. This does not have a significant impact on our results however as 94% of all connections made on all devices across all applications were over port 80 (HTTP) or port 443 (HTTPS) which we believe are very likely to represent insecure and secure connections, respectively.

B. Are applications using secure or insecure network connections?

We investigated each application's distribution of connections (*i.e.*, secure vs. insecure, depending on the port). Overall, and disregarding Securacy, 56.4% (SD=41.8%) of applications' connections are, on average, insecure (Friedman $\chi2(1)=24.59$, $p < 0.001$); Wilcoxon: $W = 728675$, $Z = -3.68$, $p < 0.001$, $r = 0.80$). However, further analysis of the top ten most frequently used applications revealed that established developers (*e.g.*, Google, Facebook) are mostly using secure connections (*i.e.*, using HTTPS – port 443, instead of HTTP – port 80). See Table 3.

Table 2. Origin-destination connection matrix (percentages).

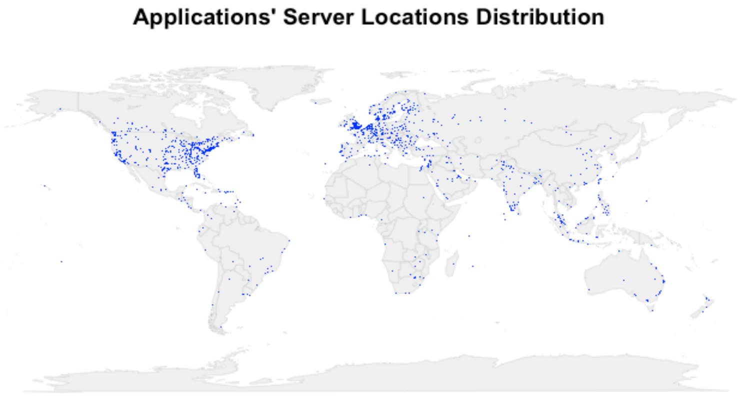
	Destination					
Origin	*Africa*	*Asia*	*Australia*	*Europe*	*N. America*	*S. America*
Africa	0	0	0	10	90	0
Asia	0	17.5	0	17.2	65.2	0
Australia	0	2.1	10.5	1.1	86.3	0
Europe	4.5	16.6	3.5	33.2	39.2	3.0
N. America	0	1.3	0.1	5.1	93.2	0.3
S. America	0	1.1	0	3.3	60	35.6

Table 3. Network security analysis of the top ten most frequently used applications.

Rank	Application	Secure (%)	Ports
1	Facebook (client)	98.2	80; 443
2	Google Play	93.1	80; 443
3	Twitter	99.3	80; 443
4	Google Mail	100.0	443
5	Google Maps	98.1	80; 443; 5228
6	Dropbox	100.0	443
7	Evernote	96.1	80; 443
8	Yahoo Mail	84.6	80; 443; 8996
9	Facebook Messenger	98.3	80; 443
10	Google Docs	100.0	443

C. How do users rate an application from a security and privacy standpoint?

We found a bimodal distribution in number of permissions participants were concerned about. **Figure 7** summarizes the data and shows that 70 out of the 218 participants did not use the feature at all. On the other extreme, 59 participants activated all



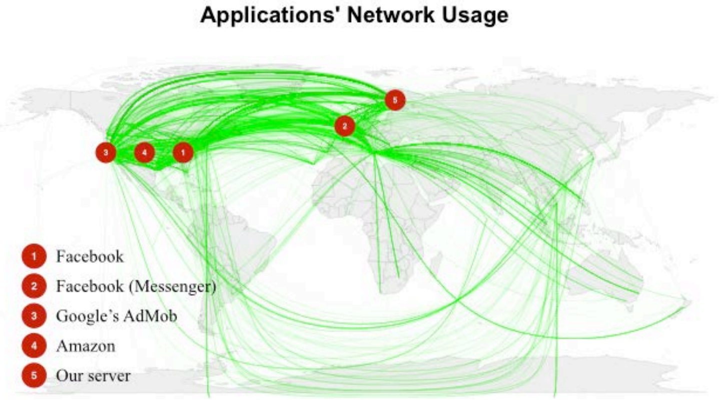


Figure 6. Global distribution of servers used from all applications on our sample (left); Visualisation of all applications' network usage. Highlighted are the top-5 most connected servers (right).

the permission concerns. Almost all participants, with four exceptions, set these only once for the duration of the study. Our participants are somewhat uncomfortable with sharing some data with applications, particularly their contact information (62.3%), profile data (61%) or access to their messages or call data (60%). On the other hand, they expressed less discomfort providing access to their browser history (37.6%), Internet connectivity (38.5%) and calendar data (45.8%)(Kruskal-Wallis $\chi^2(8)=59.74$, $p<0.001$). See **Figure 8**.

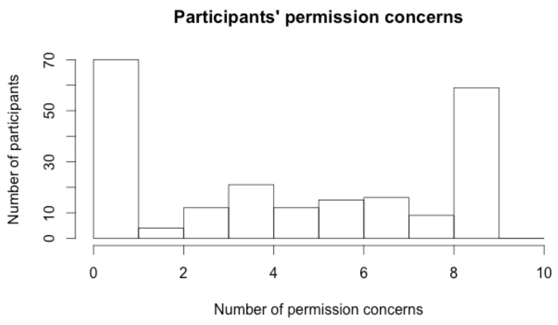


Figure 7. Participants' permission concerns distribution.

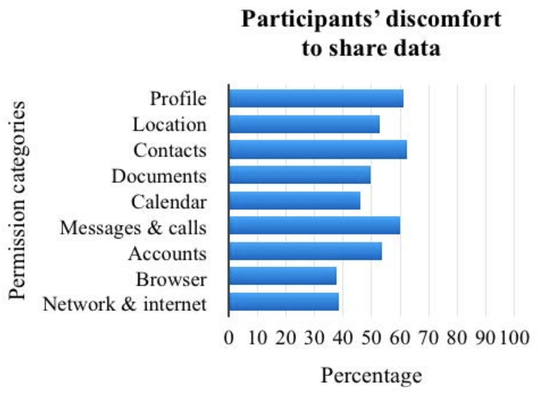


Figure 8. Participants' discomfort in sharing data with applications per category.

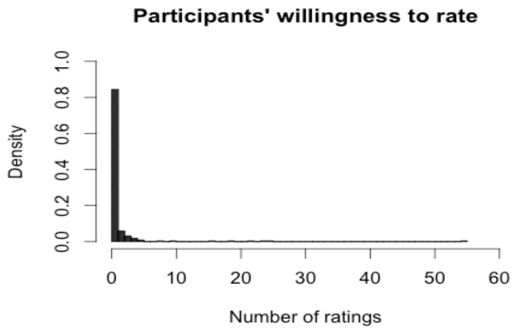


Figure 9. Probability of a participant rating an application in Securacy.

Most users do not rate applications on the Google Play store [53]. Likewise, the majority (24 out of 30, *i.e.*, 80%) of Study 1's participants stated they did not rate applications on the Google Play store. Similarly, in Study 2, the majority (184 out of 218, *i.e.*, 84.4%) of participants did not rate apps. See **Figure 9**.

However, we found that 5 out of 100 participants rated at least one application. Therefore, to rate 1,000 applications, we would hypothetically need 20,000 participants. Interestingly, on average, participants' ratings were reasonably consistent (*i.e.*, a small standard deviation of 0.24). The top five most frequently rated applications were Facebook (M=3.52; SD=1.18), Twitter (M=4.09; SD=0.79), WhatsApp (M=3.6; SD=1.28), Google+ (M=3.38; SD=1.17) and Dropbox (M=3.25; SD=0.66).

We found that *trusting the application developer* (Cockran's Q $\chi^2(2) = 96.0$, $p < 0.001$, $\phi = 0.66$) and the *application's permissions* (Cockran's Q $\chi^2(2) = 89.4$, $p < 0.001$, $\phi = 0.64$) are significant factors that are considered when rating an application. We also found that insecure connections (Cockran's Q $\chi^2(2) = 11.1$, $p < 0.001$, $\phi = 0.66$) and application's permissions (Cockran's Q $\chi^2(2) = 6.7$, $p < 0.05$, $\phi = 0.51$) are significant factors that are considered when removing an application.

Lastly, using a Kendall tau rank correlation coefficient, we investigated if a participant with more concerns (*i.e.*, higher number of active permission concerns) would, in general, give an application a lower rating. However, we did not find any correlation between the number of concerns and the applications' median rating (Kendall's $z = -1.15$, $p = 0.24$, $\tau = -0.11$).

5. DISCUSSION

Managing network security and user privacy is a challenge across multiple domains: sensor networks [11]; social networks [55]; e-learning [16]; and smart vehicles [29], to name just a few. As mobile applications increasingly connect online, it is important to understand and capture how people perceive privacy and security on their mobile phones. PeopleFinder [46] allowed users to share their location based on user-defined privacy policies and auditing. In their study, they report user difficulty in anticipating the conditions under which their peers would request access to their location. Similarly, we expected it would be challenging for casual users to anticipate all requests made by applications and pre-emptively restrict their access. In fact, Android's KitKat contained such hidden functionality: an experimental application called "*AppOps*" allowed users to disable access to sensitive permissions at runtime on a per-application basis. This feature was abandoned because, when used carelessly, it made applications non-responsive and also affected operating system performance[4], which Google feared would ultimately affect Android adoption and users experience.

We argue that requiring users to define per-app data access policies through a mechanism such as Securacy at installation time, or requiring further configuration, could also hinder participant recruitment. Alternatively, an on-demand, per-application data access strategy postpones the configuration to a when-needed basis. This model has been shown to work for iOS devices and only on rare occasions renders the applications unusable. Nauman *et al.*'s [38] work allows users to configure fine-grained allow/disallow settings for application permissions, by intercepting application data access, at some performance cost.

In contrast to previous work, in our work we leveraged anonymous feedback from users (*i.e.*, as a security rating) and data (*e.g.*, application server location and port used) to create a security and privacy *trigger* that juxtaposes user concerns on data access and application network security. This *trigger* is activated

[4] http://www.cnet.com/news/why-android-wont-be-getting-app-ops-anytime-soon/

when an application is installed or updated. More importantly, our approach is sustained by shared community interest: keeping potential malicious applications at bay. Securacy works on smartphones with Android 2.3.3 and above without any system, hardware, or software modifications or any significant performance penalty: users can simply install our app and use it. Securacy makes permission management a task that is potentially only performed once, thus reducing permission recall effort and nuisance.

5.1 Reflecting on Securacy's functionalities

Securacy was developed in approximately two months and tested intensively for two weeks before being deployed on the Google Play store. A reasonably polished application is required in order to reduce any potential risk of bad reviews, which could ultimately hinder participation [24]. The application was rated 4.4/5 –at the time of analysis – on the application store and most users reported no major application issues, instead encouraging further development and suggesting additional functionality by email or via comments on the store.

Securacy offered access to information previously inaccessible on the mobile phone: a crowdsourced mapping of mobile applications' network connections and their security status. The most popular request from participants was per-application permission management instead of a global approach. This is motivated by the fact that certain developers are perceived as worthy of access to specific permissions. For example, even though participants were uncomfortable allowing access to their contact list information, this was not be true for applications from Facebook or WhatsApp. Such functionality would also allow us to further investigate which applications and developers are perceived as trustworthy and under which circumstances and with what data.

Another request was the ability to pause or disable Securacy functionality temporarily. Some participants were concerned that, due to its ability to share server locations publicly, they could compromise their own private servers. Simply terminating the application within Android's application manager would have addressed this concern, but this is not immediately clear to users. Further development would provide users with a means of deleting a specific server or capturing server locations for specific applications.

Lastly, users also wanted to see all application ratings, not just the ones installed on their device. This would allow users to preemptively check if an application is perceived as secure or not before installation.

5.2 Sanctioned Mobile Usage

It has been three years since Felt et al.'s survey on the adequacy of Android's permissions [1]. In their paper, they reported that the majority of Android users did not pay attention to or understand permission warnings. Our findings suggest that, with a medium effect, users do review an application's permissions and have some understanding of what they mean (Cockran's Q $\chi^2(2) = 9.8$, $p = 0.007$, $\phi = 0.57$).

All of our users understood the new Android permission groups. We believe that recent news such as "*97% of mobile malware is on Android*" from F-Secure, a reputable mobile security specialist [43], and Snowden revelation in 2013 raised security and privacy awareness. In fact, in both our studies, participants were extremely apprehensive about sharing their contact information; providing access to messages and calls information; or reading their profile and user account information.

On the other hand, we found that users were willing to provide their location for rewards (*e.g.*, Verizon's phone tracking campaign [51]) or application functionality and also give applications' access to the Internet. There is a clear change in privacy concerns in recent years, as it is easier to share data with others, especially on Facebook [34]. However, an open question remains whether users' willingness to provide their location and Internet access to applications is in itself a privacy concern. We believe that the popularity of location-aware applications and the shift to cloud-based services are key to this change.

Due to Android's security design, an application cannot access device data unless the user explicitly launches the application. Therefore, it makes sense to postpone any mobile security and privacy screening to immediately after installation, not before; and only notify the user if there is cause for concern. In comparison, iOS users can install an application without notification of any required permissions. Instead, when they launch an application for the first time, they need to grant or deny access to each of the permissions requested on first use. Unfortunately on Android there is no similar permission management. It is all or nothing.

Moreover, there is also an ongoing debate regarding the utility of permissions for security and privacy protection [41], given that users do not fully understand them [1]. Recently, the extensive list of possible Android permissions (≈130 [36]) was restructured to a simpler set of permissions groups, together with a brief summary regarding the different kinds of data an application will have access to [45]. However, users must still review the list of permissions every time they install an application or update an application whose permissions have changed. This is not the primary task of the user, and consequently, users have adopted the routine of pressing the "install" button too quickly, dismissing the permissions list before they can read it [1] so they can continue with the primary task of installing the app.

We argue that the fact that users need to acknowledge a list of permissions immediately before installation of an application is disruptive. It is a barrier to the user objective of installing an application. It has been shown that only half of the time are users able to remember what a permission does [8]. Securacy makes permission management a task that is potentially only performed once, set as a user preference, and only brings permissions to the attention of the user when an application requests access to his personal data. This approach should reduce permission recall effort (remember that our participants could only recall approximately three permissions from memory) and permission nuisance, as reported by Kelley et al. [32]. After all, why does an application need to request permission to vibrate the phone [20]?

Surprisingly, we observed that 70 out of the 218 participants did not activate any of the permission concerns. Without personal permission concerns, Securacy would not notify the user, but would still display application permission usage, server locations and security, and allow the user to rate the applications. Due to participant anonymity we were unable to discuss this with them further, so we can only speculate that their motivation was interest in those additional features or simply idle curiosity.

5.3 Security Supervision

Similar to Felt *et al.* and Yan *et al.* [20, 53], only a minority of participants in both studies (20% in Study 1 and 15.6% in Study 2) rated applications. Our participants largely rely on the feedback and ratings of others. Over the six months, we received 406 ratings from 219 distinct applications, covering a very small fraction of the available applications on the market.

Moreover, given a 5% chance that a participant will rate one application, our findings suggest that we would need a larger number of participants (*e.g.*, 20,000) to rate a thousand applications. Securacy needs a much larger install base to cover majority of the applications available on the market.

With WisCom [25] and by crawling application metadata (*e.g.*, user ratings, written reviews) from the Google App Store, researchers identified three factors on how users rate and review applications: attractiveness, stability and cost. We additionally found that trust in the developer and application permissions also have a significant effect on user ratings. Similarly, we also found that permissions and insecure network usage represent a significant reason to remove an app.

We also note that participant ratings are reasonably consistent. Only a handful of application reviews may be enough to achieve consensus on a specific application's security and privacy rating. As such, it is also possible that users did not provide a rating because they agreed with the average public rating for that application from Securacy's main interface (**Figure 5 – right**). This resonates well with ProtectMyPrivacy [1], a crowdsourced security recommendation system for iOS, where just 1% of expert users provided recommendations accepted by 67.1% of the users.

5.4 Covert Network Use

Devices in North America almost exclusively (93.2%) communicate with North American servers, while 33.2% of Europe's traffic stays within Europe (**Table 2**). Given Snowden's revelations, German Chancellor Angela Merkel has encouraged Europe to vote for a network hub [5], *i.e.*, that international service providers should host their data within Europe to better protect the privacy of European citizens.

Internet access made by applications is often hidden from the user. In particular, if an application is not visible to the user (*i.e.*, a background service on Android), the user has no idea of which application is accessing the Internet and where the data is going at any given time.

Applications communicate transiently with several servers throughout the day (daily average of 115 connections, per device), mostly to those in North America. This should come as no surprise as the most popular mobile applications are often hosted in cloud-based hosting sites such as Amazon's EC2, as they are designed to scale and handle large numbers of users.

However, we find it troubling that 56.4% of the connections established by user applications were on insecure ports, a security concern. This is without considering whether applications are actually using secure connections correctly; previous studies have shown that many Android apps use SSL incorrectly [18]. Incorrectly implemented SSL can make the client vulnerable to active attacks, while not using SSL at all allows passive eavesdropping on all data. Clearly more work is needed to help developers to utilize SSL correctly [19].

In fact, the Android Developer's website warns against bypassing the devices' CA (Certificate Authority) checks, where developers implement their own (non-functional) TrustManager[5]. A valid and SSL certificate needs to issued by a trusted CA, often at a cost. This might be a cost that semi-professional developers cannot afford or are not prepared to pay.

As Edward Snowden commented in 2014 [15]: "*What last year's revelations showed us was irrefutable evidence that unencrypted communications on the Internet are no longer safe. Any communications should be encrypted by default.*" We could not agree more and a goal of our work is to raise awareness for future mobile app development.

5.5 Limitations

Although our list of permission concerns provides an overview of the types of data an application may access, it is neither exhaustive nor precise. It could be expanded in the future to capture concerns we might have missed or to improve precision. In addition, our "in-the-wild" participant sample might not be representative of the general population, as it is likely that our participants had a particular interest in the network usage, privacy and security of their applications.

6. CONCLUSION

Securacy is an Android application that reduces the burden users face when dealing with application permissions at install time. It achieves this goal by only notifying them if an application is perceived as insecure or because it requests access to permissions the user has previously recorded as concerning. Securacy builds on the fact that users do check ratings and feedback from others to assess an application. Improving the end-user management of privacy and security concerns on mobile phones is still an open challenge. To enable future work, we are making Securacy open-source[6] and release it under the Apache 2.0 license and we encourage other researchers to extend it.

7. Acknowledgements

This work is funded by the Academy of Finland (Grants 137736, 276786, 285062), TEKES (Grant 2932/31/2009), European Commission (Grants PCIG11-GA-2012-322138 and 645706-GRAGE), and the NSF award No. 1228777.

8. REFERENCES

1. Agarwal, Y. and Hall, M. ProtectMyPrivacy: Detecting and Mitigating Privacy Leaks on iOS Devices Using Crowdsourcing. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, 2013, 97-110.

2. Andrus, J., Dall, C., Hof, A., Laadan, O. and Nieh, J. Cells: A Virtual Mobile Smartphone Architecture. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles.* ACM, 2011, 173-187.

3. Backes, M., Gerling, S., Hammer, C., Maffei, M. and Styp-Rekowsky, P.V. AppGuard – Enforcing User Requirements on Android Apps. In *Tools and Algorithms for the Construction and Analysis of Systems.* Springer Berlin Heidelberg, 2013, 543-548.

4. Balebako, R., Jung, J., Lu, W., Cranor, L.F. and Nguyen, C. "Little Brothers Watching You": Raising Awareness of Data Leaks on Smartphones. In *Proceedings of the 9th Symposium on Usable Privacy and Security.* ACM, 2013, 12:1-12:11.

[5] https://developer.android.com/training/articles/security-ssl.html

[6] http://comag.oulu.fi/securacy-understanding-mobile-privacy-and-security-concerns/

5. BBC News - Can Europe go its own way on data privacy?. http://www.bbc.com/news/technology-26228176, retrieved 23/01/2015.

6. Becher, M., Freiling, F.C., Hoffmann, J., Holz, T., et al. Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices. In *Symposium on Security and Privacy.* IEEE, 2011, 96-111.

7. Benenson, Z., Gassmann, F. and Reinfelder, L. Android and iOS Users' Differences Concerning Security and Privacy. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems.* ACM, 2013, 817-822.

8. Benton, K., Camp, L.J. and Garg, V. Studying the effectiveness of android application permissions requests. In *5th Int. Workshop on Security and Social Networking.* IEEE, 2013, 291-296.

9. Beresford, A.R., Rice, A., Skehin, N. and Sohan, R. MockDroid: Trading Privacy for Application Functionality on Smartphones. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications.* ACM, 2011, 49-54.

10. Burguera, I., Zurutuza, U. and Nadjm-Tehrani, S. Crowdroid: Behavior-based Malware Detection System for Android. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices.* ACM, 2011, 15-26.

11. Chan, H. and Perrig, A. Security and privacy in sensor networks. *Computer 36,* 10, 2003, 103-105.

12. Cisco Visual Networking Index Forecast Projects 18-Fold Growth in Global Mobile Internet Data Traffic From 2011 to 2016. http://newsroom.cisco.com/release/668380/Cisco-Visual-Networking-Index-Forecast-Projects-18-Fold-Growth-in-Global-Mobile-Internet-Data-Traffic-From-2011-to-2016, retrieved 05/09/2014.

13. Consolvo, S. and Walker, M. Using the experience sampling method to evaluate ubicomp applications. *IEEE Pervasive Computing 2,* 2, 2003, 24-31.

14. Dashboards for Android Developers. https://developer.android.com/about/dashboards/index.html, retrieved 1/05/2015.

15. Edward Snowden urges professionals to encrypt client communications. http://www.theguardian.com/world/2014/jul/17/edward-snowden-professionals-encrypt-client-communications-nsa-spy, retrieved 17/09/2014.

16. El-Khatib, K., Korba, L., Xu, Y. and Yee, G. Privacy and Security in E-Learning. *International Journal of Distance Education Technologies (IJDET) 1,* 4, 2003, 1-19.

17. Enck, W., Gilbert, P., Chun, B.-G., Cox, L.P., et al. TaintDroid: An Information-flow Tracking System for Realtime Privacy Monitoring on Smartphones. In *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation.* USENIX Association, 2010, 1-6.

18. Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., et al. Why Eve and Mallory Love Android: An Analysis of Android SSL (in)Security. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security.* ACM, 2012, 50-61.

19. Fahl, S., Harbach, M., Perl, H., Koetter, M. and Smith, M. Rethinking SSL Development in an Appified World. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security.* ACM, 2013, 49-60.

20. Felt, A.P., Egelman, S. and Wagner, D. I've Got 99 Problems, but Vibration Ain'T One: A Survey of Smartphone Users' Concerns. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices.* ACM, 2012, 33-44.

21. Felt, A.P., Finifter, M., Chin, E., Hanna, S. and Wagner, D. A Survey of Mobile Malware in the Wild. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices.* ACM, 2011, 3-14.

22. Ferreira, D., Ferreira, E., Goncalves, J., Kostakos, V. and Dey, A.K. Revisiting Human-Battery Interaction with an Interactive Battery Interface. In *Ubicomp.* ACM, 2013, 563-572.

23. Ferreira, D., Kostakos, V. and Dey, A.K. AWARE: mobile context instrumentation framework. *Frontiers in ICT 2:6,* (2015) DOI: 10.3389/fict.2015.00006.

24. Ferreira, D., Kostakos, V. and Dey, A.K. Lessons Learned from Large-Scale User Studies: Using Android Market as a Source of Data. *International Journal of Mobile Human Computer Interaction 4,* 3, 2012, 28-43.

25. Fu, B., Lin, J., Li, L., Faloutsos, C., et al. Why People Hate Your App: Making Sense of User Feedback in a Mobile App Store. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2013, 1276-1284.

26. Fu, H., Yang, Y., Shingte, N., Lindqvist, J. and Gruteser, M. A Field Study of Run-Time Location Access Disclosures on Android Smartphones. In *USEC'14.* 2014.

27. Hill, R., Hansen, M. and Singh, V. Quantifying and Classifying Covert Communications on Android. *Mobile Networks and Applications 19,* 1, 2014, 79-87.

28. Hornyack, P., Han, S., Jung, J., Schechter, S. and Wetherall, D. These Aren'T the Droids You'Re Looking for: Retrofitting Android to Protect Data from Imperious Applications. In *Proceedings of the 18th ACM Conference on Computer and Communications Security.* ACM, 2011, 639-652.

29. Hubaux, J.-P., Capkun, S. and Luo, J. The security and privacy of smart vehicles. *IEEE Security & Privacy Magazine 2,* LCA-ARTICLE-2004-007, 2004, 49-55.

30. IANA TCP/IP port specifications. http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?&page=126, retrieved 14/03/2014.

31. Jing, Y., Ahn, G.-J., Zhao, Z. and Hu, H. RiskMon: Continuous and Automated Risk Assessment of Mobile Applications. In *Proceedings of the 4th ACM Conference on Data and Application Security and Privacy.* ACM, 2014, 99-110.

32. Kelley, P.G., Consolvo, S., Cranor, L.F., Jung, J., et al. A Conundrum of Permissions: Installing Applications on an Android Smartphone. In *Financial Cryptography and Data Security.* Springer Berlin Heidelberg, 2012, 68-79.

33. Kelley, P.G., Cranor, L.F. and Sadeh, N. Privacy As Part of the App Decision-making Process. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 2013, 3393-3402.

34. Kostakos, V., Venkatanathan, J., Reynolds, B., Sadeh, N., et al. Who's your best friend?: targeted privacy attacks in location-sharing social networks. In *Ubicomp.* ACM, 2011, 177-186.

35. Lin, J., Amini, S., Hong, J.I., Sadeh, N., et al. Expectation and Purpose: Understanding Users' Mental Models of Mobile App Privacy Through Crowdsourcing. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing.* ACM, 2012, 501-510.

36. Manifest permissions for Android Developers. http://developer.android.com/reference/android/Manifest.permission.html, retrieved 11/03/2014.

37. Moto X: the Google phone that's always listening. http://www.telegraph.co.uk/technology/google/10217856/Moto-X-the-Google-phone-thats-always-listening.html, retrieved 05/09/2014.

38. Nauman, M., Khan, S., Othman, A.T. and Musa, S. Realization of a user-centric, privacy preserving permission framework for Android. *Security Comm. Networks 8,* 3, 2014, 368-382.

39. Nauman, M., Khan, S. and Zhang, X. Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security.* ACM, 2010, 328-332.

40. The new Moto X is 'always listening' - and so is the NSA!. http://www.techradar.com/us/news/phone-and-communications/mobile-phones/the-new-moto-x-is-always-listening-and-so-is-the-nsa--1170553, retrieved 17/09/2014.

41. Orthacker, C., Teufl, P., Kraxberger, S., Lackner, G., et al. Android Security Permissions – Can We Trust Them? In *Security and Privacy in Mobile Information and Communication Systems.* Springer Berlin Heidelberg, 2012, 40-51.

42. Rastogi, V., Chen, Y. and Enck, W. AppsPlayground: Automatic Security Analysis of Smartphone Applications. In *Proceedings of the Third ACM Conference on Data and Application Security and Privacy.* ACM, 2013, 209-220.

43. Report: 97% Of Mobile Malware Is On Android. This Is The Easy Way You Stay Safe. http://www.forbes.com/sites/gordonkelly/2014/03/24/report-97-of-mobile-malware-is-on-android-this-is-the-easy-way-you-stay-safe/, retrieved 22/09/2014.

44. Report: NSA among worst offenders of mass surveillance, Snowden says. http://edition.cnn.com/2013/11/03/world/europe/edward-snowden-manifesto/, retrieved 08/03/2014.

45. Review app permissions. https://support.google.com/googleplay/answer/6014972?p=app_permissions&rd=1, retrieved 11/09/2014.

46. Sadeh, N., Hong, J., Cranor, L., Fette, I., et al. Understanding and Capturing People's Privacy Policies in a Mobile Social Networking Application. *Personal Ubiquitous Comput 13,* 6, 2009, 401-412.

47. Sellwood, J. and Crampton, J. Sleeping Android: The Danger of Dormant Permissions. In *Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices.* ACM, 2013, 55-66.

48. Shin, W., Kwak, S., Kiyomoto, S., Fukushima, K. and Tanaka, T. A Small But Non-negligible Flaw in the Android Permission Scheme. In *International Symposium on Policies for Distributed Systems and Networks.* IEEE, 2010, 107-110.

49. Truong, H.T.T., Lagerspetz, E., Nurmi, P., Oliner, A.J., et al. The Company You Keep: Mobile Malware Infection Rates and Inexpensive Risk Indicators. In *Proceedings of the 23rd International Conference on World Wide Web.* International World Wide Web Conferences Steering Committee, 2014, 39-50.

50. Tu, G.-H., Peng, C., Li, C.-Y., Ma, X., et al. Accounting for Roaming Users on Mobile Data Access: Issues and Root Causes. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, 2013, 305-318.

51. Verizon's offer: Let us track you, get free stuff. http://money.cnn.com/2014/07/22/technology/mobile/verizon-tracking/index.html?hpt=hp_t2, retrieved 22/07/2014.

52. Wahlberg, T., Paakkola, P., Wieser, C., Laakso, M. and Roning, J. Kepler - Raising Browser Security Awareness. In *IEEE 6th Int. Software Testing, Verification and Validation Workshops (ICSTW),* 2013, 435-440.

53. Yan, B. and Chen, G. AppJoy: personalized mobile application discovery. In *MobiSys.* New York, New York, USA, 2011, 113-126.

54. Yang, Z., Yang, M., Zhang, Y., Gu, G., et al. AppIntent: analyzing sensitive data transmission in android for privacy leakage detection. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security.* ACM, 2013, 1043-1054.

55. Zhang, C., Sun, J., Zhu, X. and Fang, Y. Privacy and security for online social networks: challenges and opportunities. *Network, IEEE 24,* 4, 2010, 13-18.

56. Zhang, Y., Yang, M., Xu, B., Yang, Z., et al. Vetting Undesirable Behaviors in Android Apps with Permission Use Analysis. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security.* ACM, 2013, 611-622.