Third best student paper award "Study on privacy of parental control applications"

Álvaro Feal Fajardo
IMDEA Networks / IMDEA Software
Madrid, Spain
alvaro.feal@imdea.org

***Abstract*—Parental control applications are one kind of mobile software programs used by parents to monitor and control the use that their kids make of their cellphone. In this study we aim to learn about the privacy of these applications. We combine manual analysis of the Android marketplace with dynamic analysis of the application itself to learn about the ecosystem of these apps and their privacy problems. We studied 17 different applications of which 10 where fully analyzed using dynamic analysis. We found 11 privacy issues in 8 of these 10 apps. We also found that 17% of the permissions requested by these apps are not clearly mapped to any app functionality and that 50% of the applications do not clearly explain their communication model to users.**

***Index Terms*—Privacy, Android, Parental control applications**

Contribution type *Third best student paper award (extended abstract)*[1]

I. Parental Control Apps

Parental control apps are used by parents to keep track of their child's activities on the internet. Usually, parents install the application in the kid's phone and can later configure a set of rules to dictate what their child can and cannot do. The privacy threats on these type of applications are huge, not only because they gather all kinds of private data, but also because these data are sent over the Internet towards third party servers. Also, the number of downloads on Google Play for the apps in our study shows that these types of apps are widely used. Nevertheless, to the bests of our knowledge there has not been a previous study on this type of mobile applications.

A. Threat model

Due to this communication model, there are several points where privacy problems may occur and lead to private data leaks. When sending information from the phone to the Internet, communications must be secure. Otherwise, an observer on the network could intercept the communication and gain access to private data. Therefore, all communications that include private data must be encrypted using TLS. The other way privacy leaks may occur is by the mishandling of these data by the controller of the software. If they do not treat data correctly in any of the points were these data is handled, then privacy of the user is compromised.

B. Understanding the ecosystem

We study a shortlist of 17 apps using different measurements:

Google Play Store data allows us to learn that these apps have a number of installs that revolves around 100.000 to 500.000 unique downloads. The mean user rating is around 3.4 out of 5.

License and pricing wise, there are three main business models used by these apps. Free, meaning that the whole app is free to use; Premium, where all of the features are pay to use; and Freemium, where there is a core of free features but users that pay get access to more features.

Parental control features allow us to separate these apps in blocking and monitoring apps. The first type only does active blocking of some of the phone's features, while the latter also keeps track of phone activity in order to show this info to parents.

Our shortlist consist of *14 monitoring apps* which we will further analyze, plus *3 blocking apps* for completeness.

II. Communication model

This type of applications are installed on the children's phone. Nevertheless, parents can later access data logs and blocking settings from other devices, such as their own phone or any device with a web browser. This behavior tells us that this type of applications must send private data to Internet server, where it will be stored, processed and later accessed by parents. The problem with this model is that when the app description does not clearly explain this model, non expert users such as parents will not know that their child's data is leaving their phone and going through the Internet. We study the descriptions provided in Google Play for the 14 monitoring apps in our shortlist. For this, we manually check if the developers clearly explain this behavior to parents, by searching in the text description for words such as "server", "Internet communication" or "sending of information". Our results show that only 50% of the studied applications clearly explain that private data will leave the phone.

III. Permissions study

Overprivilege in Android applications is an issue that has been broadly studied. [1][2][3]. Applications tend to request more permissions than those necessary for the correct functioning of the software. In most of the cases this is due to lack of understanding of the Android permission, reusing code found in Internet tutorials, unfinished features and requesting permissions for related methods to the ones that are being used. It is safe to say than in most cases, an unnecessary permission request means a developer error and not a malicious attempt to get access to phone resources.

Table I
Percentage of permissions that can be mapped to the functionality of the app

Necessary	Maybe necessary	Unnecessary
83%	3.5%	13.5%

[1]*This works is the result of my Master thesis presented in the Universidad Politécnica de Madrid and supervised by Manuel Carro and Carmela Troncoso*

We study the permissions requested by the 14 monitoring applications and manually match this to their advertised features and to the runtime behavior. When there is a clear correlation between a requested permission and one functionality, we mark this as "Clear". When these correlation is not clear or we can't be sure if the permissions is necessary we tag this as "Maybe necessary". Finally, when a permission is clearly unmatched with any feature, we mark it as "Unnecessary". Our results show that 11 of the 14 apps request permissions that are not necessary. Table I shows the results of our study, where out of all the permissions requested for the applications, only 83% are clearly necessary.

IV. Privacy threats

To learn about the privacy problems present in these apps, we dynamically analyze them using *TaintDroid*. This is a tweaked version of Android that identifies private information during runtime and tags it. This tag is attached to the data during the flow of the execution, and if it ever reaches a data sink (i.e., network interface and phone storage), TaintDroid shows that the tagged data has been leaked. We run TaintDroid in an Android emulator, which leads to only 7 of the 14 monitoring apps to work. Therefore, this analysis has been applied to 7 monitoring apps and 3 blocking apps.

A. Private data mishandling

To study these apps, we individually install each one in the TaintDroid emulator and follow a set of actions that aim to trigger data leakage. These actions are: installation and setup, Internet browsing (whitelisted and blacklisted sites), app usage (blocked and allowed apps), calling a phone number, sending a SMS and setting a fake location for the phone. Whenever we find that data is sent to a domain, we study this domain to know if it belongs to a third party service or to the developers. We also save the IP address where the info is sent to analyze network traffic (see Subsection IV-B). After testing the 7 monitoring apps we found that all of them presented at least one of these problems:

Leakage of information before policy acceptance: Users must read and accept the privacy policy before the app can gather any type of information. Nevertheless we found one app that never shows the privacy policy to the user and two apps that gather information before the user has even created an account (which means implicit acceptance of the policy).

Leakage of information prior to permission grant: When an user gives an application permission to access certain information, the info collection should start at that moment. We found two applications that request access to the SMS and call logs and send a complete history, including data previous to the permission being granted. We believe that this is a problem in the Android permission API as well, because such an action should be blocked.

Sending of sensitive information to third parties: We consider that data that allows a party to learn sensitive information about the user (e.g, SMS and call logs, browsing history and location data) should not be shared with third parties. We found two applications that sent such data to third parties, one of them sends every URL visited by the child while the other one shares location data.

B. Insecure communications

We inspect the network traffic between each one of the apps and the controller servers, as well as any communication that might happen with third party services. To do so, we gather the TCP dumps of the phone's network communications and use the tool *Wireshark* to check if the communications are encrypted. We are able to filter the incumbent communications by using the IPs gathered in the previous subsection.

We found that 3 of the 7 monitoring apps send private information in the clear. Even more important is that two of those are actually sending every URL visited by the child in the clear, making it possible to infer the complete browser history. In the case of blocking apps, we don't expect them to send information over the Internet, because the parent manages the blocking from the child's phone, making the communication model different (and safer). One of the studied apps sends the IMEI code (unique identifier of a phone) unencrypted.

Table II shows a summary of the results found in the ten analyzed applications.

Table II
Number of apps where each of the privacy threats were found

Privacy threat	N^o of apps
Private data sent to 3rd party	2
Leakage of info prior to permission grant	2
Sending of private data before policy acceptance	3
Sending of private information without encryption	4

V. Ethical considerations and future work

We want to address that we have not harvested real data from users at any point of our experiments. Instead, we have created fake accounts with traffic generated by ourselves to avoid the gathering of data from real users. We also want to address that given the limitations spotted in this work, we are currently working on extending our method to be able to be more robust on finding dissemination of private data. We will do a responsible disclosure before publishing our final work.

VI. Conclusions

To the best of our knowledge, this is the first privacy study on parental control applications. These set of apps are highly intrusive making their study an important issue.

We analyzed 14 monitoring applications (7 of them partially) and 3 blocking apps and found that 7 out of 14 monitoring apps do not clearly explain their communication model, 17% of the requested permissions for 7 of the monitoring apps are not clearly necessary and that there are 11 total privacy issues between one of the blocking apps and the 7 monitoring apps.

References

[1] Felt, Adrienne Porter et al. "Android permissions demystified." ACM *Conference on Computer and Communications Security*, 2011.

[2] Felt, Adrienne Porter et al. "Android permissions: user attention, comprehension, and behavior." *Symposium on Usable Privacy and Security*, 2012.

[3] Backes, Michael et al. "On Demystifying the Android Application Framework: Re-Visiting Android Permission Specification Analysis." *USENIX Security Symposium*, 2016.

[4] Enck, William and Gilbert et al. "TaintDroid: An Information-flow Tracking System for Realtime Privacy Monitoring on Smartphones." *USENIX OSDI*,2010