IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 5 Issue 4, April 2018
ISSN (Online) 2348 – 7968
www.ijiset.com

AUTOPPG: A Literature Survey on Automatic Generation of Privacy Policy for Android Applications

Jihan C [1] and Aneesh M. Haneef [2]

[1] Department Of Computer Science and Engineering, MES College Of Engineering, Kuttippuram

[2]Asst. Professor Department Of Computer Science and Engineering, MES College Of Engineering, Kuttippuram

Abstract

A privacy policy is a statement informing users how their information will be collected, used, and disclosed. Failing to provide a correct privacy policy may result in a fine. However, writing privacy policy is tedious and error-prone, because the author may not understand the source code well as it could have been written by others (e.g., outsourcing), or the author does not know the internal working of third-party libraries used. In this paper, we propose and develop a novel system named AutoPPG to automatically construct correct and readable descriptions to facilitate the generation of privacy policy for Android applications (i.e., apps). Given an app, AutoPPG first conducts static code analysis to characterize its behaviors related to users' personal information, and then applies natural language processing (NLP) techniques to generating correct and accessible sentences for describing these behaviors. The experimental results using real apps and crowd sourcing indicate that: (1) AutoPPG creates correct and easy-to-understand descriptions for privacy policies; (2) the privacy policies constructed by AutoPPG usually reveal more operations related to users' personal information than existing privacy policies.

1. Introduction

As smart phones have become an indispensable part of our daily lives, users are increasingly concerned about the privacy issues of the personal information collected by apps. Although the Android system lists all permissions required by an app before its installation, such approach may not help users understand the app's behaviors, especially those related to users' personal information, due to the lack of precise and accessible descriptions.

Actually, the Federal Trade Commission (FTC) suggested mobile developers to prepare privacy policies for their apps. writing privacy policy is tedious and error-prone because of many reasons invoked. Moreover, the developer may not know the internals
of the integrated third-party libraries, which usually do not provide source code. Existing approaches for automatically generating privacy policies cannot solve these issues because they rely on human. For example, the author of a privacy policy may not well understand the app's source code, which could be outsourced, or the precise operation of each API
intervention, such as answering questions like "what personal information do you collect?", and few of
them analyze code. It is worth noting that the tool Privacy Informer could only generate privacy policies for apps created by App Inventor instead of normal apps.

2. Modules Of AutoPPG

AUTOPPG consists of three modules:
2.1: Document analysis module. Given an API and its description from the corresponding API document,this module identifies the personal information used by the API automatically by leveraging the Google Java Style and employing information extraction techniques . The output of this module is used by the static code analysis module to determine the personal information collected by an app.

```
Input: desc: API description, name_method: name entity in
       method name, name_class: last part of class name
Output: personal information used in this API
desc_tree = StanfordParser_Tree(desc);
desc_dept = StanfordParser_Dept(desc);
root = ExtractRoot(desc_dept);
obj = ExtractObj(desc_dept, root);
nameInfo = null;
if Exist(name_method) then
    nameInfo = name_method;
else
    nameInfo = name_class;
end
if nameInfo != null then
    simValue = Similarity(obj, nameInfo);
    if simValue > threshold then
        return obj;
    else
        ConatinName=obj.contain(nameInfo);
        if ContainName == true then
            return obj;
        else
            info=FindInfo(desc_tree, desc_dept);
            obj = obj + info;
            return obj;
        end
    end
else
    return obj;
end
```

2.2 : Static code analysis module . Given an app without source code and the mapping of selected APIs to the personal information from the document analysis module, this module disassembles the app's APK file, turns its statements into intermediate representation (IR), and inspects the IRs to profile the app by performing the following four steps: (1) finding the personal information collected by apps;(2) locating the collector of the personal information identified in (1); (3) determining whether or not the app asks for users' consent explicitly before invoking the selected APIs. identifying the app's information retention strategy.

```
Input: stmt: statement in code
api = ExtractCalledAPI(stmt);
sensitive = SensitiveAnalysis(api);
reachable = ReachabilityAnalysis(api);
if sensitive == true ∧ reachable == true then
    Info = MapApiToInformation(api);
    User = UserIdentification(api);
    Conditions = ConditionExtraction(api);
    RetatinOrNot = RetentionAnalysis(api);
    TransferredOrNot = TransferAnalysis(api)
    return (Info, User, Condition, RetainOrN
    TransferredOrNot);
end
return null;
```

2.3: Privacy policy generation module. Taking in an app's profile identified through static code analysis, this module aims at generating readadable format.

3. Proposed System Design

- AutoPPG, a novel system that automatically constructs correct and readable descriptions to facilitate the generation of privacy policy for Android apps. To our best knowledge, AutoPPG is the first system that can construct such information by leveraging static codeanalysis and NLP techniques.

- Tackle several challenging issues in developing AutoPPG, including automatically mapping APIs to personal information, profiling an app's behaviors related to personal information, and constructing correct and readable descriptions. These techniques can also be applied to solve other research problems, such as malware detection.
- implement AutoPPG and perform careful experiments with real apps and crowd sourcing to evaluate its performance. The experimental results show that AutoPPG can construct correct and easy-to-understand descriptions for privacy policy. Actually, the privacy policies resulted from AutoPPG usually reveal more apps' behaviors related to users' personal information than existing privacy policies. Moreover, most developers, who reply us, would like to use AutoPPG to facilitate them.

4. Conclusion

AUTOPPG is a noval system implemented for android applications. It automatically constructs, corrects readable descriptions to facilitate the generation of privacy policy. Various approaches for AUTOPPG is described. Through comparative study found out that the AUTOPPG is the most efficient one with high performance, and more secure than other systems. Problems of the existing systems were identified. Modifications are brought in to increase the accuracy of the system. Moreover AutoPPG is more readable than existing privacy policy.

References

[1] L. Yu, X. Luo, X. Liu, and T. Zhang, "Can We Trust the Privacy Policies of Android Apps?," in Proc. IFIP/IEEE DSN, 2016.

[2] T. Breaux, A. Anton, "Analysing regulatory rules for privacy and security requirements,"IEEE Transactions through android framework}, vol. 293, no. 10, 2008.

[3] Le Yu. Tao Zhang. "AUTOPPG: Automatic generation of privacy policy,"(IEEE transactions on Information Forensics and Security, Vol. 14 (8),2017.

First Author Presently pursing Post Graduation in Computer Science and Technology. Completed Bachelor Degree in Computer Science and Engineering in 2015 from MES College Of Engineering Kuttippuram.

Second Author Working as Assistant Professor in Computer science and Engineering Department, MES College of Engineering, Kuttippuram, India.