Android Permissions Demystified
paper summary

Ruturaj Kiran Vaidya
University of Kansas

1 Summary

Android enforces install time permissions per application. User must approve the permissions specified in the manifest before the installation of that application. Install time permissions provide more security and help to reduce bugs. But, developers sometimes request more permissions than required, making install time permission system ineffective. Authors claim that this happens because of lack of information and inconsistencies in android documentation. So, they empirically determined Android 2.2's access control policy. At first, they generated test cases for API calls (using Randoop and their own custom tool), content providers and intents. Second, Their testing resulted into a permission map, "that identifies what permissions are needed for api calls"[1]. Based on that, they characterized, how permissions checks are disstributed through api. "Using automated testing techniques, they achieved 85% coverage of android API"[1]. They discovered 1259 methods with permission checks, but the android 2.2 documentation has only 78. They also discovered 6 inconsistencies in the documentation. To analyze the application and detect the over-privilege, they designed the tool, called Stowaway, which consists of two parts - "a static analysis tool that determines what api calls an application makes, and a permission map that identifies what permissions are needed for each API call"[1]. The tool identified 35% applications having unnecessary privileges.

2 Contribution

The authors contributed by introducing a tool (Stoaway). This tool detects the overprivileged android applications. They constructed a permission map, which can be useful in further research in this area. They recognized the patterns of developer error which lead to overpriviledge. Also, Using automated tesing, they determined android's security access control policy[1].

3 Strengths

The paper gives a really good explanation about android's permission system background, permission enforcement, etc. They also pointed out inconsistencies and incompleteness of android official documentation. Their testing reveals permission requirements for 1259 methods (a 16 times improvement over the official android documentation).

4 Limitations

Handling Java reflection is important, but paper fell short in that area (maybe due to user input or environmental variable dependencies).

5 Future Work

More research can be done, to achieve 100% coverage of android api. Also, as authors claim, more research can be done in resolving Java reflective calls.

References

[1] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, David Wagner, *Android Permissions Demystified*, CCS, (2011)