PermDroid: Handling over-privileged Android applications based on the minimum permissions set identification

Doctorant (email) : mohammed-el-amin.tebib@univ-grenoble-alpes.fr
Encadrants de Thèse : Oum-El-Kheir Aktouf (LCIS- Grenoble INP), Pascal André (LS2N - Univ. De Nantes), Mariem Graa (IMT Atlantique)
Laboratoire : LCIS, Grenoble INP, Université Grenoble Alpes

1 Context "Android Applications Access Control: Permissions"

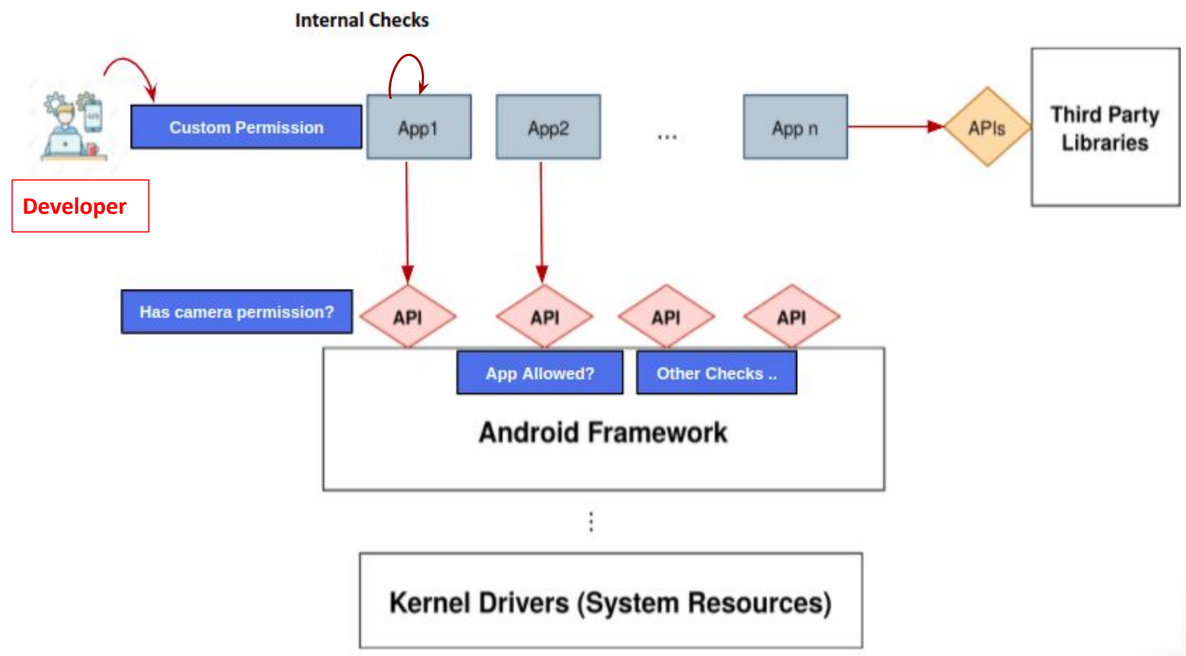
1.1 Definition

Permissions are authorizations declared/defined by developers in the application source code

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
   <permission android:name="android.permission.CAMERA"/>
   <permission android:name="android.permission.INTERNET"/>
   <permission android:name="android.permission.ACCESS_NETWORK_STATE"/>

   <application>  ... </application>
</manifest>
```

1.2 Where are permissions used?



2 PermDroid Motivation: Preventing Over-privileged Apps

Developers mistakenly implement over-privileged apps: Permissions Used > Permissions Declared

2.1 How could developers detect over-privileged apps?

Algorithm 1 Detecting Over-privileged Applications

```
declaredPerms ← getDeclaredPerms(manifest);
apiCalls ← getApiCall(sourceCodeFiles);
Initialize perm.used = false forallperm ∈ declaredPerms
for each apiCall ∈ apiCalls do
    perms ← getPermissionsOfApiCall(apiCall);
    for each p ∈ perms do
        if p ∈ declaredPerms then perm.used=true;
        end if
    end for
end for
```

2.2 Review of existing solutions

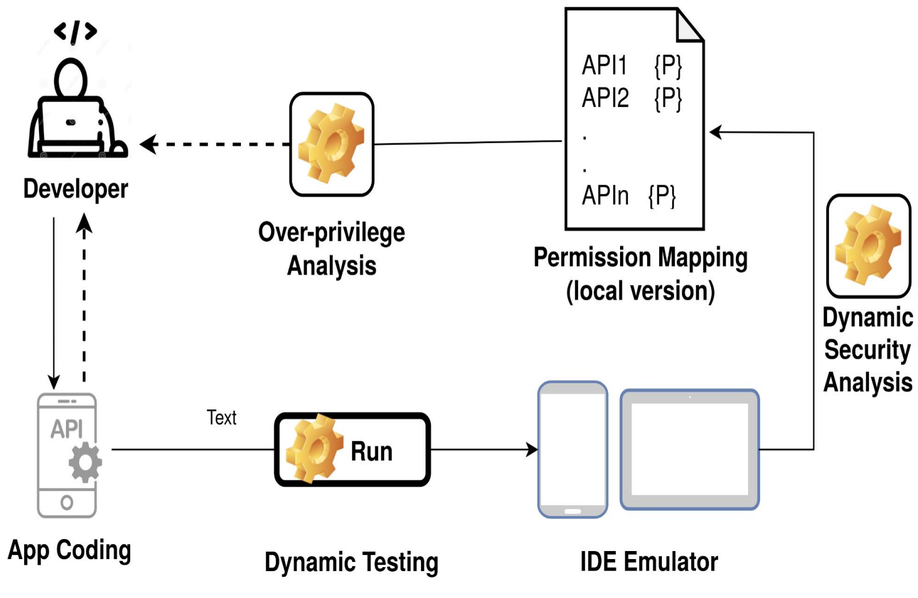
Tool	Android API Level	Permission Mapping	Analysis Approach	Availability
PerHelper	12	Pscout 2011	Static	No
PermitMe	12	Pscout 2011	Static	No
Curbing	9	Manual	Static	No
PermDroid	9..31	Pscout 2011, Arcade 2018, Dynmo 2021	Static; Dynamic	Yes

2.3 Challenges

- **Uncomplete static analysis approaches: Java Reflection, Native Code, etc.**
- **Availability. None of these tools is available to be used in real projects**
- **Outdated existing solutions, due to the evolution of permissions and APIs.**

3 Proposal: A collaborative hybrid analysis approach

3.1 On-IDE: Combined Static & Dynamic Analysis



Completeness

Automatic Updates

3.2 Collaborative Running Process