Naturwissenschaftlich Technische Fakultät

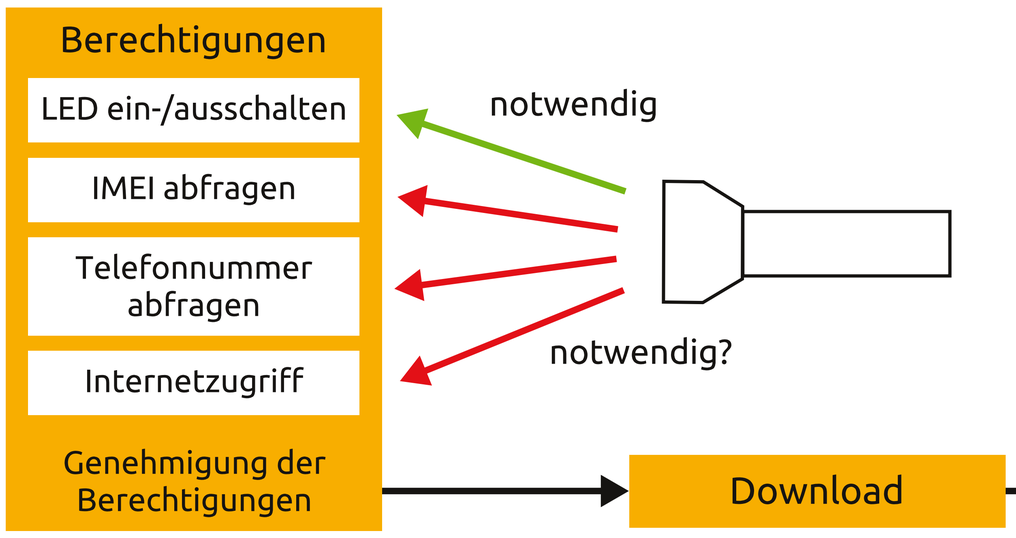
UNIVERSITÄT SIEGEN

Statische Daten- und Informationsflussanalyse von Android-Apps

Markus Schmidt

Ist mein Smartphone sicher?

Welche Berechtigungen braucht eine **Taschenlampen-App** für ihre Funktion?



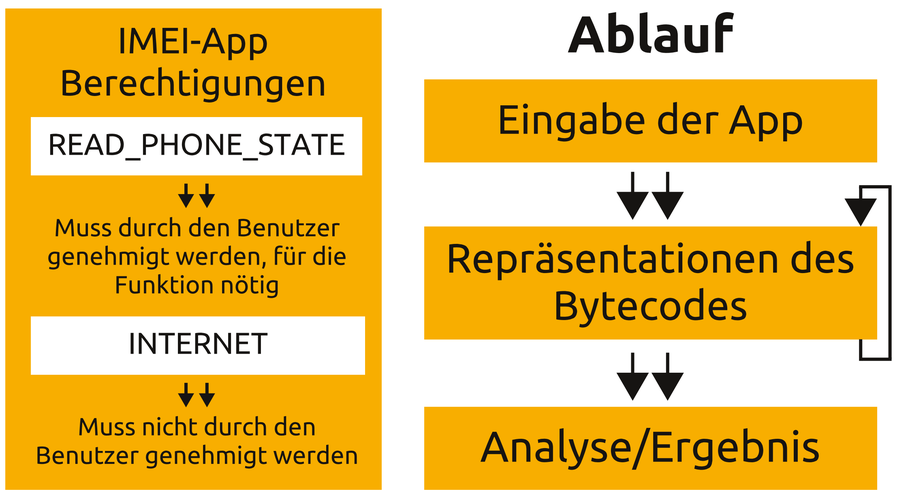
Zeigt eine App schadhaftes Verhalten?

Als Anwender von Programmen oder Apps lässt sich oft schwierig nachvollziehen, was Apps mit den eingegebenen Daten machen. Baut das Programm eine Verbindung zu einem Server auf und überträgt Daten, die für die Kernfunktion irrelevant sind? Anhand der angeforderten Berechtigungen lässt sich keine Aussage über die Schadhaftigkeit treffen. Die Datenflussanalyse untersucht den Verlauf des Inhalts der Variablen, während die Informationsflussanalyse den Datenfluss inhaltlich beurteilt. Die Analyse erfolgt ohne Kenntnis des Quelltextes auf Basis einer Zwischenrepräsentation des Bytecodes.

Überprüfung vor der Installation/Ausführung einer App: Methoden der statischen Analyse

Analyse der Prozeduren einer Beispiel-App

Zum Test wurde eine App zur Anzeige der IMEI entwickelt, die mithilfe eines Buttons die IMEI des Smartphones anzeigt.

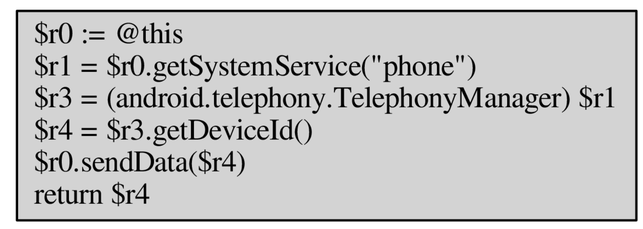


Ablauf

Quelltext zur Evaluation:

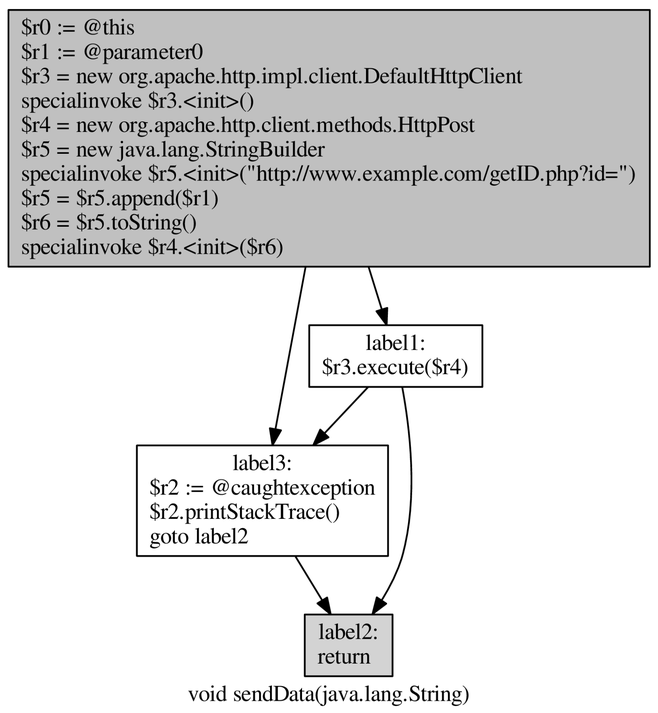
```
public String ShowIdOnClick() {
   HttpClient httpClient = new DefaultHttpClient();
   String theId = mngr.getDeviceId();
   sendData(theId);
   return theId;
}

public void sendData(String data) {
   TelephonyManager mngr =
   (TelephonyManager)getSystemService(Context.TELEPHONY_SERVICE);
   HttpPost request =
   new HttpPost("http://www.example.com/getID.php?id=" + data);
   try {
   httpClient.execute(request);
   } catch (Exception e) {
   e.printStackTrace();}
}
```

Datenflussanalyse des Bytecodes mit dem Framework Soot[1]



java.lang.String ShowIdOnClick()

Soot[1] rekonstruiert die Funktion ShowIdOnClick() mit dem Rückgabewert String. Die hier gewählte Repräsentation ist ein Kontrollflussgraph, der den Datenfluss erkennen lässt. Die Funktion liefert nicht nur die IMEI zurück, sondern ruft auch sendData($r4) auf.



void sendData(java.lang.String)

Aus dem Kontrollflussgraphen der Funktion sendData (String a) ist ersichtlich, dass mit dem Parameter $r1 ($r4 aus ShowIdOnClick()) ein PHP-Skript aufgerufen wird, welches die IMEI empfängt. Am Ende stehen die URL und IMEI in $r5. In label1 wird der Aufruf gestartet.

Die durch Soot generierten Kontrollflussgraphen zeigen die Funktionen, welche im Quelltext zu sehen sind. Die untersuchten Methoden* führen zumeist auch die Informationsflussanalyse durch und finden daher mögliche Datenschutzverletzungen automatisiert.

*Quellen

- [1] Ondrej Lothak und Laurie Hendren Patrick Lam, Eric Bodden. The Soot framework for Java program analysis: a retrospective. (Soot)
- [2] Patrick McDaniel und Swarat Chaudhuri William Enck, Damien Octeau. A Study of Android Application Security. (ded decompiler)
- [3] Kwangkeun Yi und Junbum Shin Jinyung Kim, Yongho Yoon. ScanDal: Static Analyzer for Detecting Privacy Leaks in Android Applications. (ScanDal)
- [4] Barbara G. Ryder Karim O. Elish, Danfeng (Daphne) Yao und Xuxian Jiang. A Static Assurance Analysis of Android Applications. (user-intention program dependence analysis)
- [5] Siegfried Rasthofer, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves le Traon, Damien Octeau, Patrick McDaniel, Christian Fritz, Steven Arzt. Highly precise taint analysis for android applications. (FlowDroid)

Ergebnis

Die statische Daten- und Informationsflussanalyse ist ein wichtiges und effektives Werkzeug bei der Erkennung von Apps mit schadhaftem Verhalten vor der Installation/Ausführung. Die Analysemethoden sind trotz unterschiedlicher Vorgehensweisen genau. Eine absolute Aussage kann mit keiner Methode getroffen werden[2]. Es werden hauptsächlich unnötige Abfragen des Standortes und der eindeutigen IMEI, IMSI und der Telefonnummer von schadhaften Apps durchgeführt[2345].

Markus Schmidt
Universität Siegen
Lehrstuhl für Betriebssysteme und verteilte Systeme
markus1.schmidt@student.uni-siegen.de