Taint Analyse für Android Apps

Thomas Czogalik

Betreuer: Simon Greiner

1 Motivation

Durch fehlerhafte oder absichtlich manipulierte Programme können Datenlecks enstehen und diese ausgenutzt werden. In Abbildung 1 1 sehen wir ein Ausschnit aus einem Java Programm. Es wird zunächst eine Datenbank Verbindung hergestellt und im Anschluß soll ein SQL Statement ausgeführt werden. Wie wir in Zeile 6 sehen, hängt das Statement von einer Benutzereingabe aus Zeile 5 ab. Ein Angreifer könnte in diesem Fall durch folgende Eingabe: "foo; DROP TABLE users" die Datenbank users löschen. Er könnte aber auch mit einem SELECT Statement Daten aus der Datenbank holen. Solche Szenarien lassen sich durch Überprüfung vermeiden. So ein Fehler ist aber nicht immer so einfach zu erkenne und wird übersehen. Da Computersysteme heutzutage in nahezu allen Bereichen unseres Lebens intigriert sind, können solche Datenlecks fatale folgen haben. Besonders Smartphones verwalten und verarbeiten viele vertauliche und private Daten und komunizieren dabei meißt mit der Außenwelt. Im Februar 2015 befanden sich im Google Play Store ca. 1.4 Millionen Apps. Diese sind jedem zugänglich, der auf seinem mobilen Gerät das Betriebssystem Android installiert hat. Bei so einer großen Anzahl Apps bietet der Google Play Store eine große Angriffsfläche. Deshalb ist es notwendig, den Fluss sensitiver Daten nachvollziehen zu können. Dies ist mithilfe einer Taint Analyse möglich.

```
1 connection = …
2 stmt = connection.createStatement();
3
4 BufferedReader br = …
5 String name = br.readLine();
6 String sql = "SELECT * FROM users WHERE name=" + name + ";";
7
8 stmt.executeQuery(sql);
```

Fig. 1. test caption

2 Taint Analyse

2.1 statisch vs. dynamisch

Bei der Tain Analyse unterscheidet man zwischen statischer und dynamischer Taint Analyse. Der Vorteil der statischen Variante ist, dass das Kompilat nicht ausgeführt werden muss. Dies ist Hilfreich, da heutige Malware erkennen kann ob sie überwacht wird und kann ihr Verhalten anpassen. Im folgenden wird sich auf die statische Taint Analyse beschränkt.

2.2 Spezifikation und Vorgehen

Die Idee der Taint Analyse ist, dass jede von außen veränderbare Variable ein Sicherheitsrisiko birgt. Ihr Ziel ist es die Software gegen externe Angriffe sowie interne Risiken abzusichern. Dazu sucht die Taint Analyse nach Datenflüssen von möglichen tainted sources zu einem sink. Als source wird eine Funktion bezeichnet, die Quelle sensitiver Daten ist. Ein sink ist eine Funktion, die Daten möglicherweise an nicht vertrauenswürdige Beobachter weitergibt. Welche Funktionen im einzelnen sources und sinks sind, muss vor der Taint Analyse angegeben werden.

2.3 Formal

Im Folgendem wird eine Formalisierung der Taint Analyse vorgestellt mithilfe von Schlussregeln.