Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-SIT

Seidlgasse 22 / 9, 1030 Wien
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

Inffeldgasse 16a, 8010 Graz
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

DVR: 1035461
ZVR: 948166612
UID: ATU60778947

HANDBUCH FÜR APPLIKATION ZUR ERKENNUNG VON TIMING-UNTERSCHIEDEN IN DER ANDROID API

Version 1.0 vom 08.06.2020

Gerald Palfinger – gerald.palfinger@iaik.tugraz.at

Inhaltsverzeichnis

Inhaltsverzeichnis 1
1. Einleitung 1
2. Systemvoraussetzungen 1
3. Installation 1
3.1. Kompilieren von Source Code 1
4. Inbetriebnahme 1
4.1. Konfiguration 1
4.2. Starten 2
5. Handhabung 2
6. Lizenz 2

1. Einleitung

Das bereitgestellte Erkennungstool ermöglicht es, Timing-Schwachstellen in Methoden der Android API zu erkennen. Dazu sucht es Methoden, deren Ausführungszeit stark variiert bei Verwendung unterschiedlicher Parameter.

2. Systemvoraussetzungen

Zur Ausführung der Software wird ein Smartphone mit Android 10 oder neuer benötigt.

3. Installation

3.1. Kompilieren von Source Code

Der bereitgestellte Source Code kann mit Android Studio kompiliert werden. Die Applikation kann direkt aus Android Studio installiert werden oder als APK auf das Smartphone übertragen werden.

4. Inbetriebnahme

4.1. Konfiguration

In der `parameter_values.yaml` Konfigurationsdatei können Parameter vordefiniert werden. Diese befindet sich im Ordner `app/src/main/res/raw/`. Parameter können auf drei verschiedenen Ebenen konfiguriert werden; entweder für alle Methoden in der ganzen API, für alle Methoden einer Klasse oder für eine spezifische Methode. Dies funktioniert wie folgt:

Um einen Wert für alle Methoden der API zu definieren wird folgender Syntax verwendet:

```
<Parametername>: <value>
Beispiel:
pids: [10795]
```

Um einen Wert für alle Methoden einer Klasse zu definieren wird folgender Syntax verwendet:

```
<Klassenname>:
  "*":
    <Parametername>: <value>
Beispiel:
java.security:
  "*":
    type: "AndroidKeyStore"
```

Um einen Wert für eine spezifische Methoden einer Klasse zu definieren wird folgender Syntax verwendet:

```
<Klassenname>:
  <Methodenname>:
    <Parametername>: <value>
Beispiel:
android.hardware.input.InputManager:
  getInputDevice:
    id: 0
```

Nach Änderung der Datei muss die Applikation neu kompiliert und installiert werden.

4.2. Starten

Das Erkennungstool kann durch Klick auf das Icon am Smartphone gestartet werden.

5. Handhabung

Die Erkennung von Timing-Schwachstellen wird durch das Klicken des Buttons in der Applikation gestartet. Die Resultate werden im Ordner der Applikation gespeichert (`/storage/emulated/0/Android/at.asit.project.timing/`).

6. Lizenz

Das bereitgestellte Projekt wird unter den Bedingungen der Open-Source-Lizenz für die Europäische Union (EUPL) V1.1 bereitgestellt (siehe Lizenz.pdf).