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 CODE INJECTION TOOLS

Version 1.0 vom 09.08.2019
Gerald Palfinger – gerald.palfinger@iaik.tugraz.at

Inhaltsverzeichnis

1. Einleitung

Um mobile Cross-Plattform Applikationen zu entwickeln werden oft Frameworks eingesetzt, die auf JavaScript und HTML5 basieren. Dadurch bringen solche Applikationen jedoch nicht nur den Vorteil der Plattformunabhängigkeit mit sich, sondern auch die Schwachstellen von Browseranwendungen, allem voran die Möglichkeit, potentiell schadhaften Code einzuschleusen. In dieser Dokumentation wird ein Tool vorgestellt, das die Verwendung der vor Code Injection schützenden Browserfunktion Content Security Policy (CSP) überprüfen und beurteilen kann (im weiteren Content Security Policy-Evaluator genannt). Ebenso wird ein weiteres Tool vorgestellt, welches feststellen kann, ob in eine Applikation potentiell Code über Apache Cordova-Plugins eingeschleust werden kann (im weiteren Code Injection-Erkennungstool genannt).

2. Systemvoraussetzungen

Zur Ausführung der Tools wird ein Java Development Kit benötigt. Durch die Abhängigkeit der verwendeten Bibliotheken wird dieses in der Version 8 vorausgesetzt. Für die Ausführung des Content Security Policy-Evaluators wird Python 3 benötigt. Zur manuellen Erstellung der Test-Applikation werden die Apache Cordova-Kommandozeilenapplikation benötigt[1]. Alle weiteren benötigten Bibliotheken werden mitgeliefert.

3. Installation

Zur Installation die Zip-Datei mit einem geeigneten Programm entpacken.

[1] Installation siehe https://cordova.apache.org/#getstarted

3.1. Kompilieren von Source Code

Der Content Security Policy-Evaluator kann wie im folgenden Kapitel beschrieben direkt verwendet werden. Zur Ausführung des Code Injection-Erkennungstools wird IntelliJ empfohlen. Ein Projekt mit den benötigten Einstellungen wird mitgeliefert.

4. Inbetriebnahme

4.1. Konfiguration

4.1.1. Content Security Policy-Evaluator

Die zu untersuchenden Android Packages sollten in den Ordner `APKs` gespeichert werden.

4.1.2. Code Injection-Erkennungstool

Das zu untersuchende Android Package sollte im `resources`-Ordner des Evaluators gespeichert werden.

4.2. Starten

4.2.1. Content Security Policy-Evaluator

Um den Evaluator zu starten, sollte die Datei `evaluate_csp.py` mit Python 3 auf der Konsole ausgeführt werden. Daraufhin werden alle Android Packages im Ordner `APKs` untersucht.

4.2.2. Code Injection-Erkennungstool

Um das Erkennungstool zu starten, sollte die Klasse `AnalyserMain` gestartet werden. Als Argument wird der Name des zu untersuchenden Android Packages erwartet.

4.3. Testen

Zur Überprüfung der Funktionalität des Code Injection-Erkennungstools wurde eine Demo-App erstellt, welche anfällig für Code Injection ist. Diese Applikation befindet sich im Ordner `test_app`. Ein fertiges Android Package dieser Applikation zur Evaluierung des Erkennungstools wird mitgeliefert. Alternativ kann das Android Package auch mit folgendem Kommando erstellt werden:

```
cordova build android
```

Das erstellte Android Package kann dann wie oben beschrieben mit dem Erkennungstool verwendet werden.

5. Lizenz

Die verwendeten Bibliotheken WALA und DASCA stehen unter der Eclipse Public License 2.0.