Thema für Bachelorarbeit / Masterarbeit
Javascript Code-Restriktion und Sandbox

Das Problem

Gelegentlich möchte man Code aus unbekannter Quelle auf seinem Browser laufen lassen. Bindet man Javascript aus unbekannter Quelle in seine Webseite ein, so ist der Rechner des Betrachters durch die Browser-Sandbox vor vielen Schadfunktionen des fremden Javascript-Codes geschützt. So kann der Javascript Code etwa nicht auf beliebige Dateien des Endgeräts zugreifen.

Bei serverseitigem Javascript (konkret: Node.js) besteht ein solcher Schutz hingegen nicht. Der Benutzer muss dem in die Server-seitige Anwendung eingebundenen Modul vollständig vertrauen. Da diese Situation unbefriedigend ist, wurden verschiedene Formen von Javascript-Sandboxen für Node.js entwickelt. Diese sollen weniger vertrauenswürdigen Code unter eingeschränkten Rechten ausführen. Leider sind viele dieser Sandbox Lösungen aber nicht zuverlässig. So finden sich etwa zu den Modulen gf3, vm und vm2 im Internet etliche erfolgreiche Angriffe, wie man die Sandbox durchbrechen kann. Die Nutzung einer als sicher angepriesenen Sandbox bleibt daher auf absehbare Zeit noch ein riskantes Unternehmen.

Neben dem Sandbox-Ansatz gibt es aber noch andere Möglichkeiten. So kann man den auszuführenden Javascript Code analysieren und feststellen, ob er bestimmte "verbotene" Funktionen enthält. Diese können auf eine Fehlermeldung in der Code Analyse führen, in welchem Fall man den Code eben nicht ausführt.

Eine weitere wichtige Code-Restriktion, die man öfters benötigt, ist die Freiheit von Seiteneffekten (sogenannte pure functions). Diese werden bei verschiedenen Frameworks (etwa: flux/redux) und bei bestimmten Template Engines benötigt.

A□fgabenstell□ng

Je nach Art der Arbeit (Bachelor, Master, Projekt), nach Interesse und Vorbildung des Bearbeiters sollen im Rahmen dieser Arbeit einige der folgenden Fragen beantwortet werden. Eine Bearbeitung von verbundenen Themen durch ein Team ist ebenso denkbar. Die genaue Fragestellung definieren wir in der Vorbesprechung.

Es soll ein Code-Analysator entwickelt werden, der Javascript Code parst und anschließend die Einhaltung bestimmter **Code-Restriktionen** überprüft, etwa die Freiheit von Aufrufen von Systemfunktionen oder die Einbindung bestimmter externer Module.

Es soll ein Code-Analysator entwickelt werden, der Javascript auf die **Freiheit von Seiteneffekten** untersucht.

Für Seiteneffekt-freie Ausdrücke von Javascript soll schließlich ein partieller Evaluator für "pure functional Javascript" entwickelt werden.

Es erweist sich dabei von besonderem Vorteil, dass für Javascript gute, in Javascript selber geschriebene Parser zur Verfügung stehen (Beispiel: Esprima).

Hinweis: Natürlich kann man durch die Analyse des Programms nicht entscheiden, ob der Code Seiteneffekte hat oder ob er bestimmte Funktionen aufruft (siehe Theorie-Vorlesung). Man kann aber überprüfen, ob ein bestimmter Programmierstil eingehalten wird, der garantiert, dass der Code frei von Seiteneffekten ist bzw. bestimmte Funktionen nicht aufruft. Das Ziel der Arbeit widerspricht also nicht den Theoremen von Rice und Shapiro.

Ansprechpartner: Prof. Clemens Cap