Forschungsprojektarbeit des Profils Information and Cyber Security

Fuzzing mit Token-Learning

Ausgangslage

Fuzzing ist eine sehr beliebte Methode, um Fehler und Sicherheitslücken in Software aufzuspüren. Dabei werden zufällig generierte Eingaben erstellt, die auf gültigen Beispiel-Eingaben basieren, um zu beobachten, wie das Computerprogramm diese Eingaben verarbeitet. Ziel ist es, Eingaben mit kleinen Modifikationen zu verwenden, um Randfälle auszulösen, die das Programm zum Absturz bringen könnten. Fuzzing hat sich als äusserst effizient erwiesen, insbesondere in Kombination mit Grammatiken oder Tokens, um besser passende Eingaben für ein Programm zu erzeugen. Allerdings kann das Definieren einer Grammatik für Eingaben schwierig sein und erfordert viel Vorbereitungszeit, wobei eine Automatisierung oft nicht möglich ist. Tokens stellen einen alternativen Ansatz dar, der einfacher ist, aber bei dem Automatisierungsmethoden noch nicht umfassend erprobt wurden.

Zielsetzung / Methodik / Vorgehen

In diesem Projekt soll das Fuzzing auf Basis von Tokens analysiert werden. Ein Token ist ein spezifisches Muster einer Eingabe und hilft dabei, bessere Eingaben für den Fuzzing-Prozess zu generieren. Tokens müssen jedoch vom Benutzer vor dem Fuzzing-Prozess definiert werden. Eine bessere Lösung wäre es, wenn Tokens während des Fuzzing-Prozesses basierend auf dem Feedback des Fuzzers generiert würden. Dies kann zwei Vorteile haben: Erstens ist es einfacher, den Fuzzer auszuführen, ohne die Tokens im Voraus definieren zu müssen, und zweitens könnte der Fuzzer Tokens finden, die für den Benutzer nicht offensichtlich sind.
Als Grundlage für die Implementierung eines Token-Lernmechanismus kann libAFL qemu verwendet werden.
Ziel dieses Projekts ist es, einen Token-Lernmechanismus zu entwickeln und zu implementieren und diesen anschliessend mit geeigneter Software zu evaluieren. Wenn Sicherheitslücken gefunden werden, sollen diese im Rahmen einer «Responsible Disclosure» dem Hersteller gemeldet bzw. CVEs registriert werden.

Teilaufgaben

Das Projekt kann in 2 bis 3 Stufen durchgeführt werden: Fortlaufende Projekte 7 (Token-Lernmechanismus), 8 (PoC Implementierung) und 9 (Realisierung mit libAFL qemu, Testcase und Evaluation)

Studienart: [x] Vollzeitstudium
[x] Teilzeitstudium 50%

Projektorganisation: Einzelarbeit

Projektfinanzierung: Keine

Arbeitsort: Windisch

Advisor: Dr. Christopher Scherb