Bachelor-/Masterarbeit

Effiziente Ermittlung der Code-Abdeckung bei Fuzzing-Tests für Software-Anwendungen

Ziel

Fuzzing ist eine Technik zur Entdeckung von Schwachstellen in Softwareanwendungen. Während eines Fuzzing-Laufs werden gezielt fehlerhafte Daten an eine Anwendung übergeben, um zu prüfen, ob die Implementierung diese Daten verarbeiten kann oder ob es zu Abstürzen oder unerwarteten Fehlerzuständen kommt.

Um ein möglichst umfassendes Bild davon zu erhalten, ob ein Großteil der Anwendung erfolgreich getestet wurde, werden Daten zur sogenannten *Code Coverage* erhoben. Diese zeigen, welche Stellen im Code durch den Fuzzer überhaupt erreicht wurden. Ziel ist es, eine möglichst hohe Abdeckung durch die Tests zu erreichen.

Viele Fuzzing-Tools führen dazu eine Instrumentierung des Source Codes durch. Durch gezieltes Einfügen von Messcode während des Kompilierens können Informationen zur Code Coverage erhoben werden. Das setzt jedoch voraus, dass der Quellcode verfügbar ist und mit einem Compiler kompiliert werden kann, der eine solche Instrumentierung unterstützt.

Security Researcher und Penetration Tester haben jedoch nicht immer Zugriff auf den Quellcode. Um auch ohne Quellcode aussagekräftige Ergebnisse zur Code-Abdeckung zu erhalten, kann *Dynamic Binary Instrumentation* (DBI) eingesetzt werden. Mit DBI wird ein Programm zur Laufzeit instrumentiert, um zu überprüfen, ob bestimmter Code erreicht wird. Diese Möglichkeit ist jedoch in vielen verfügbaren Fuzzern nicht vorhanden.

Ziel dieser Arbeit ist die Entwicklung eines Werkzeugs zur Erhebung von Code-Abdeckungsstatistiken während eines Fuzzing-Laufs. Ein eigenständiges Tool soll entwickelt werden, das ein gängiges DBI-Framework verwendet, um ein gegebenes Programm zu instrumentieren. Das Werkzeug soll Informationen zur Code-Abdeckung erfassen und in einem geeigneten Format für die Weiterverarbeitung speichern.

Die korrekte Funktionsweise des Werkzeugs wird anhand verschiedener Software-Anwendungen untersucht. Hierfür wird ein vorhandener Fuzzer für das Fuzzing verwendet, während die zu testende Software parallel mit dem neu entwickelten Werkzeug instrumentiert wird.

Anforderungen und Voraussetzungen

- Einarbeitung in das Thema der Dynamic Binary Instrumentation (DBI)
- Untersuchung und Auswahl gängiger DBI-Frameworks für Eignung zur Erhebung von Code Coverage
- Entwicklung eines Tools zur Instrumentierung mit Hilfe des ausgewählten Frameworks
- Auswertung anhand verschiedener Beispielanwendungen (Fuzzer ist vorhanden)
- Voraussetzungen: Kenntnisse der Sprache Python und Bereitschaft sich in das Thema einzuarbeiten

Ansprechpartner

Prof. Dr. Lothar Braun | ✉ lothar.braun@tha.de | ✆ +49 8215586-3378

THA_innos

Das Institut für innovative Sicherheit (THA_innos) bietet eine Vielzahl von Abschluss- und Projektarbeiten im Themenfeld der Cyber Security an. Unser Team unterstützt Studierende dabei mit Know-How und Praxiserfahrung und ist zudem offen für eigene Themenvorschläge. Durch die enge Zusammenarbeit mit der Forschungsgruppe vor Ort im MRM-Gebäude lernen Studierende sowohl das Institutsleben, als auch die aktuelle Forschung von THA_innos kennen.

Weitere Informationen auch unter `https://innos.tha.de`