Forschungsprojektarbeit des Profils Information and Cyber Security

Entwicklung eines Hardware-basierten Fuzzing-Frameworks für Embedded und Industrial Control Systems

Ausgangslage

Eingebettete Systeme und Industriesteuerungen bilden das Rückgrat kritischer Infrastrukturen in zahlreichen Branchen wie Fertigung, Automobilindustrie, Energieversorgung und Medizintechnik. Die Sicherheitsanalyse dieser Systeme stellt eine besondere Herausforderung dar, da traditionelle Softwaretestmethoden oft nicht anwendbar sind. Dies liegt an proprietären Betriebssystemen, fehlenden Debug-Schnittstellen oder der Notwendigkeit, den laufenden Betrieb nicht zu stören. In der Schweiz mit ihrer starken Präsenz in den Bereichen Präzisionstechnik, Automatisierung und Medizintechnologie sind solche Systeme weit verbreitet. Trotz fortschrittlicher Fuzzing-Techniken für Desktop- und Serveranwendungen fehlen spezialisierte Lösungen für eingebettete Systeme, die ohne Quellcodezugriff auskommen und dennoch detaillierte Einblicke in die Systemreaktion ermöglichen. Das Monitoring über Hardware-Schnittstellen wie JTAG oder mittels Power-Analyse durch Logic Analyzer bietet einen vielversprechenden, nicht-invasiven Ansatz, der bisher jedoch nicht systematisch für Fuzzing-Zwecke genutzt wird.

Zielsetzung / Methodik / Vorgehen

Ziel dieses Projekts ist die Entwicklung eines neuartigen Fuzzing-Frameworks, das speziell auf eingebettete und industrielle Steuerungssysteme ausgerichtet ist und Hardware-basierte Tracing-Methoden nutzt. Im ersten Schritt soll eine umfassende Analyse verschiedener Tracing-Möglichkeiten über JTAG-Schnittstellen und Logic Analyzer erfolgen, wobei besonders die Erfassung von Ausführungspfaden und Seitenkanaleffekten (z.B. Stromverbrauchsmuster) betrachtet wird. Darauf aufbauend wird ein adaptives Fuzzing-System konzipiert, das Eingabedaten über geeignete Schnittstellen (serielle Verbindungen, Feldbusse, Netzwerk) an das Zielsystem sendet und gleichzeitig die Systemreaktion über die Hardware-Schnittstellen überwacht. Ein besonderer Schwerpunkt liegt auf der Entwicklung von Feedback-Mechanismen, die aus den Trace-Daten und Power-Analyse-Ergebnissen Rückschlüsse auf die Programmausführung ziehen können, ohne dass Instrumentierung des Zielsystems nötig ist. Diese Informationen werden dann genutzt, um den Fuzzing-Algorithmus zu steuern und die Testabdeckung zu optimieren. Für die praktische Umsetzung wird eine modulare Testumgebung aufgebaut, die verschiedene eingebettete Systeme und industrielle Steuerungen unterstützt und flexibel an unterschiedliche Hardware-Debugging-Interfaces angepasst werden kann. Die Lösung soll in der Lage sein, auch ohne detaillierte Systemkenntnisse potenzielle Schwachstellen wie Pufferüberläufe, Timing-Probleme oder unzureichende Eingabevalidierung zu identifizieren.
Verlangte Skills: Kenntnisse in Embedded-Programmierung, Programmierung (Python, C/C++, Rust)

Teilaufgaben

Das Projekt kann in 3 Phasen durchgeführt werden: Fortlaufende Projekte 7 (Analyse der Tracing-Möglichkeiten über JTAG und Logic Analyzer, Konzeption des Fuzzing-Frameworks), 8 (Implementierung der Hardware-Anbindung, Entwicklung der Feedback-Mechanismen basierend auf Trace- und Power-Analysedaten) und 9 (Integration in ein vollständiges Fuzzing-Framework, Evaluation an verschiedenen industriellen und eingebetteten Systemen, Dokumentation und Veröffentlichung)

Studienart: [ ] Vollzeitstudium
[ ] Teilzeitstudium 50%

Projektorganisation: Einzelarbeit (im Umfeld eines Projektteams)

Projektfinanzierung: -

Arbeitsort: Windisch

Advisor: Prof. Dr. Christopher Scherb