

Softwareprojekt: Iridium

ein Analyseframework für kompilierte Programme

Inokentiy Babushkin

Abteigymnasium Brauweiler

11.05.2015 / Colloquium zur Besonderen Lernleistung

Inhalt

Grundlagen

- Kurze Einführung in CPU-Internia
- Reverse Engineering

Features

- Prinzipien
- Kontrollflussanalyse
- Analyse von Daten auf dem Stack
- Analyse optimisierter Integerdivisionen

Inhalt

Grundlagen

Kurze Einführung in CPU-Internia

Reverse Engineering

Features

Prinzipien

Kontrollflussanalyse

Analyse von Daten auf dem Stack

Analyse optimisierter Integerdivisionen

- ▶ Die CPU unterstützt einzelne Instruktionen, darunter auch Sprünge
- ▶ Diese werden durch sog. Opcodes im Speicher dargestellt
- ▶ Können auch in Form von Assembler-Befehlen beschrieben werden
- ▶ Vom Abstraktionsgrad kaum mit Hochsprachen zu vergleichen

Inhalt

Grundlagen

Kurze Einführung in CPU-Internia

Reverse Engineering

Features

Prinzipien

Kontrollflussanalyse

Analyse von Daten auf dem Stack

Analyse optimisierter Integerdivisionen

Definition hier: Wiederherstellung von Hochsprachencode aus kompilierten Programmen

- ▶ Komplexer und zeitaufwändiger Prozess, da
 - ▶ Software zunehmend komplex
 - ▶ Compiler den Code optimieren
- ▶ Sehr nützlich
 - ▶ Malware-Analyse
 - ▶ Wiederherstellung von altem Programmcode
 - ▶ Entwicklung von Exploits

(Halb-)Automatisierte Ansätze

- ▶ Decompiler
- ▶ Analyseframeworks

Inhalt

Grundlagen

Kurze Einführung in CPU-Internia
Reverse Engineering

Features

Prinzipien

Kontrollflussanalyse

Analyse von Daten auf dem Stack

Analyse optimisierter Integerdivisionen

Erweiterbarkeit

- ▶ Modularer Aufbau
- ▶ Sinnvolles API-Design
- ▶ Multiple Eingabeformate
- ▶ Lesbarkeitsorientierter Code

Freie Software

- ▶ Verwendung der GPLv3
- ▶ Codequalität und Funktionsumfang durch Community unterstützt

Inhalt

Grundlagen

Kurze Einführung in CPU-Internia
Reverse Engineering

Features

Prinzipien

Kontrollflussanalyse

Analyse von Daten auf dem Stack

Analyse optimisierter Integerdivisionen

Analyse anhand von Kontrollflussgraphen

- ▶ Schrittbasierter Reduktionsalgorithmus für einzelne Funktionen
- ▶ Unterscheidung aller wichtigen Hochsprachen-Programmelemente
- ▶ Strukturierte Darstellung

Inhalt

Grundlagen

Kurze Einführung in CPU-Internia
Reverse Engineering

Features

Prinzipien
Kontrollflussanalyse
Analyse von Daten auf dem Stack
Analyse optimisierter Integerdivisionen

Heuristiken zur Untersuchung lokaler Variablen

- ▶ Untersuchung des Stackframes der jeweiligen Funktion
- ▶ (Teilweise) Erkennung von Arrays, Datentypen und Pointern

Inhalt

Grundlagen

Kurze Einführung in CPU-Internia
Reverse Engineering

Features

Prinzipien
Kontrollflussanalyse
Analyse von Daten auf dem Stack
Analyse optimisierter Integerdivisionen

Optimierte Integerdivision

- ▶ Von modernen Compiler bei konstantem Divisor angewendet
- ▶ Teure Divisions-Instruktion wird durch eine Reihe von Additionen, Shifts und Multiplikationen ersetzt
- ▶ Für Menschen keine Rückschlüsse auf den Divisor möglich, deswegen Implementation eines entsprechenden Algorithmus

Zusammenfassung

- ▶ Reverse-Engineering und Analyse kompilierter Programme sind häufig notwendig, allerdings meist auch schwierig und mit enormem Aufwand verbunden.
- ▶ Gleichzeitig sind die automatisierten Lösungsansätze für diese Aufgaben nicht oder nur kaum für interaktive Arbeit geeignet, weswegen mit dem vorliegenden Projekt ein entsprechender Prototyp vorgelegt wird.
- ▶ Ausblick
 - ▶ Stabilisierung des Codes
 - ▶ Erweiterung um Eingabemodule
 - ▶ Weitere Features