ATIS - Software Reverse Engineering

Prof. Dr. Christian Dietrich
dietrich@internet-sicherheit.de

2 Lab: Static Code Analysis

Aufgabe

Das Ziel dieser Praktikumsaufgabe besteht darin, sich mit Disassemblierung für die x86-Architektur vertraut zu machen. Dazu soll ein gegebenes 32-bit x86-Binaerprogramm unter Zuhilfenahme eines Disassembling Toolkits disassembliert werden. Als Eingabe dient ein Programmausschnitt in Binärform sowie die Information, dass der Eingabecode zur Laufzeit an die Speicheradresse 0x100030d1 gemapped wird. Die Ausgabe ist die Disassembly in tabellarischer Form, mindestens bestehend aus folgenden Spalten:

- Speicheradresse der Instruktion
- Hexadezimaldarstellung einer Maschinencode-Instruktion
- Disassembly einer Maschinencode-Instruktion (Mnemonic und Operanden)

Zur Verdeutlichung sei als Beispiel folgender Code in Hexadezimalform gegeben sowie in Binärform in der Datei mit dem Namen `binarycode_example`:

```
55 8B EC 51
```

Die resultierende Disassembly sieht wie folgt aus:

```
0x100030d1: 55                  push   ebp
0x100030d2: 8bec                mov    ebp, esp
0x100030d4: 51                  push   ecx
```

Eingaben

- Datei mit dem Namen `binarycode1`
- Speicheradresse 0x100030d1; an diese Adresse wird der Code aus der Datei gemapped

Abzuliefernde Ergebnisse

Disassembly in tabellarischer Textform wie oben beschrieben.

Hinweise

- Als Eingabe dient ein Programmausschnitt in Binärform, der in der Datei mit dem Namen `binarycode1` vorliegt. Diesen Code sollen Sie disassemblieren.
- Die SREVM enthält den Disassembler **capstone** (`www.capstone-engine.org`), der u.a. aus Python und Java heraus angesprochen werden kann. Sie können capstone (oder auch jeden beliebigen anderen Disassembler) für die Disassemblierung benutzen. Hinweise zur Programmierung mit capstone in Python sind unter `http://www.capstone-engine.org/lang_python.html` zu finden, für Java unter `http://www.capstone-engine.org/lang_java.html`.