Prof. Dr. Michael Eichberg
michael.eicherg@dhbw-mannheim.de

DHBW
Duale Hochschule
Baden-Württemberg
Mannheim

Reverse Engineering

Ziel dieser Übung ist es einen ersten Einblick in den Bereich des Software Reverse Engineering zu bekommen. Da es beim Reverse Engineering fast immer darum geht Schutzsysteme zu umgehen, ist dies auch in diesem Fall Ihre Aufgabe.

Für diese Aufgaben ist es (höchstwahrscheinlich) notwendig (ein bisschen) Code zu schreiben. Für die Abgabe können Sie ggf. Ihren Code in folgenden Sprachen abgeben: Java, Python, Scala, C, C++, Rust, JavaScript, Prolog, Bash, Zsh.

Die beiden Aufgaben ermöglichen ganz verschiedene Lösungen. Der Code der ersten sollte aber schneller/einfacher verständlich sein. Dies bedeutet aber nicht zwangsläufig, dass die Lösung einfacher ist.

Es ist im Rahmen des Reverse Engineering immer so, dass viele Wege zum Ziel führen. Sollte ein Weg nicht gehen, dann probieren Sie einen anderen.

Das Wichtigste ist Beharrlichkeit!

1. `simplesecurepp` [Vorlesung - Live Demo]

Ihnen liegt eine verschlüsselte Datei vor (42.enc), die mit dem Program simplesecurepp verschlüsselt wurde. Informationen zum verwendeten Passwort liegen nicht vor. Es ist jedoch davon auszugehen, dass das Passwort sicher ist. Ziel ist die Entschlüsselung der Datei.

Analysieren Sie die Anwendung, um einen möglichen Ansatzpunkt zu finden, um die Daten erfolgreich zu entschlüsseln.

Sollten Sie Code schreiben in Hinblick auf einen Bruce Force, dann sollten Sie diesen ggf. auch parallelisieren, um möglichst schnell zum Ergebnis zu gelangen.

2. `securepp` [Übung]

Entschlüsseln Sie die Datei Poem.enc, die mit dem Program securepp verschlüsselt wurde.

Sie können zum Beispiel das Programm über ein Shellscript oder ein anderes Java Programm ansprechen, um zu versuchen das Passwort zu Bruteforcen. Alternativ oder ergänzend können Sie versuchen mittels Reverse Engineering herauszufinden wie die Verschlüsselung implementiert wurde und ob sich daraus effizientere/andere Möglichkeiten ergeben. Selbstverständlich können Sie versuchen die Entschlüsselung ggf. auch in (z. B.) Python nachbauen.

Die Password-Policy, die dem Passwort zugrunde liegt ist die folgende:

- mind. 16 Zeichen
- mind. 4 Ziffern
- mind. 2 Großbuchstaben
- mind. 2 Kleinbuchstaben
- mind. 4 verschiedene Sonderzeichen aus dem Zeichensatz: -!$?#@
- Es dürfen nicht alle Sonderzeichen hintereinander vorkommen; d. h. es muss mindestens zwei Blöcke von Sonderzeichen geben; ein Block kann ein oder mehrere Sonderzeichen enthalten.

Darüber hinaus sei Ihnen bekannt, dass die Person in seinen Passworten sehr gerne die Namen der Orte verwendet, die er bereits bereist hat. Ihnen sei außerdem bekannt, dass die Person zwischen einer großen Anzahl von Großstädten in Westeuropa unterwegs war. Darüber hinaus trennt er die Namen der Städte in bekannten Passwörtern häufig mit “.”, “_" oder “-“. Weiterhin sei Ihnen bekannt, dass er oft das Datum bzw. Jahr verwendet an denen er die Städte in den letzten Jahren besucht hat. Ansonsten verhält er sich augenscheinlich recht typisch was das Anfügen von Zahlen und Sonderzeichen betrifft. Beides ist in bestehenden Passwörtern am häufigsten in entsprechenden Blöcken am Ende zu finden.