Hochschule Esslingen
University of Applied Sciences

Studienarbeit

Reverse Engineering mit Ghidra

im Studiengang Softwaretechnik und Medieninformatik
der Fakultät Informationstechnik
Sommersemester 2021

Stefan Schanz

Zeitraum: 16.03.2021 - 01.09.2021
Prüfer: Prof. Dr. Rer. Nat. Tobias Heer
Zweitprüfer: Prof. Dr. Dominik Schoop

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Esslingen, den 2. September 2021 S. Sanz

Unterschrift

Inhaltsverzeichnis

1 Einleitung

In dieser Studienarbeit wurde die Thematik des Reverse Engineerings aufgefasst. Es sollte ein Kurs anhand des Tools 'Ghidra' entworfen werden. In diesem Kurs werden verschiedene grundlegenden Themen eingeführt. Dazu gehört ein Überblick über das Thema des Reverse Engineering, wobei verschiedene Techniken und Anwendungsmöglichkeiten erläutert werden. Zudem wird der Mechanismus erklärt, wie eine Binärdatei entsteht. Weitere Kapitel beschäftigen sich mit den Programmiersprachen C und Assembler, deren Verständnis grundlegend für das Reverse Engineering ist. Außerdem soll ein Überblick über verschiedene Linux Kommandozeilen Tools und verschiedene Reverse Engineering Tools geschaffen werden.

Zudem wurden für den Kurs einige Übungsaufgaben entworfen. Diese sind mit dem jeweiligen Wissensstand des behandelten Kapitels abgestimmt. Zudem wird anhand der Übungsaufgaben in das Tool 'Ghidra' eingeführt. Alle Übungsaufgaben (bis auf die Übungsaufgaben des Kapitels mit C und Assembler) können mit dem Tool 'Ghidra' bearbeitet werden.

Dieses Dokument ist eher als Beschreibung des Arbeitsprozesses für die Erstellung des Kurses gedacht. Inhalt hierbei soll die Struktur des Kurses und die Beschreibung der Notwendigkeit bestimmter Kapitel sein.

2 Ziele und Motivation

Das Ziel der Studienarbeit ist das Erstellen eines Kurses für andere Studierende. Der Kurs soll in die Thematik des Reverse Engineerings mit Hilfe des Tools 'Ghidra' einführen. Hierbei handelt es sich um eine Einführung in den Bereich des statischen Reverse Engineering.

Dabei soll eine breite Grundbasis des Reverse Engineerings geschaffen werden.

Ein weiteres Ziel der Studienarbeit ist, dass interessierte Studierende sich mit der Thematik auseinandersetzen können und nahezu alle relevanten Grundlagen innerhalb des Kurses beigebracht werden. Es ist nicht unbedingt notwendig für die Bewältigung des Kurses andere Informationsquellen heranzuziehen. Der Kurs ist als eigenständiger und vollständiger Kurs gedacht. Allerdings werden grundlegende Programmierkenntnisse für das Reverse Engineering vorausgesetzt.

Die Motivation der Ausarbeitung eines Kurses in diesem Themenbereich liegt darin, dass die Hochschule (zumindest im Moment) kein Modul in diesem Bereich anbietet. Diese Lücke soll zumindest teilweise durch diesen Kurs gefüllt werden.

Zudem ist es durch das Ausarbeiten des Kurses möglich, sich umfangreiches Wissen in diesem Bereich anzueignen. Vor der Ausarbeitung des Kurses konnte man sich zwar vorstellen, wie das Reverse Engineering aussieht, aber es war nicht möglich, dies ohne Vorkenntnisse zu meistern. Genau diese Vorkenntnisse und noch vieles mehr konnten durch die Ausarbeitung des Kurses gesammelt werden.

3 Struktur des Kurses

Die Struktur des Kurses ist folgendermaßen aufgebaut. Es werden Kapitel für die Theorie und für die Praxis bereitgestellt. Beispielsweise handeln diese Kapitel von allgemeineren Informationen im Bezug zum Thema Reverse Engineering oder erklären grundlegende Funktionsweisen, wie das Kapitel 'Vom Code zum Binary'.

Die praktischen Kapitel beinhalten Übungsaufgaben, die auf den jeweiligen theoretischen Kapiteln aufbauen. Teilweise bieten die Übungsaufgabenteile Informationen über die Verwendung von 'Ghidra' an, die bei der Lösung der Aufgaben helfen. Anderweitig werden Tipps gegeben, falls bei dem jeweiligen Aufgabenteilen Schwierigkeiten auftreten.

Die Struktur des Kurses sieht folgendermaßen aus:

1. Was ist Reverse Engineering?
    a) Erklärung Reverse Engineering
    b) Reverse Engineering im Bereich der IT
    c) Beispiel Reverse Engineering
    d) System Level Reverse Engineering
    e) Code Level Reverse Engineering
2. Überblick
    a) Anwendungsmöglichkeiten für Reverse Engineering
    b) Unterschiedliche Techniken des Reverse Engineering
3. Rechtliches
4. Übungsaufgaben (Teil 1)
5. Vom Code zur Binärdatei
    a) Preprocessing
    b) Compilation
    c) Assembly
    d) Linking
6. Systemkomponenten
    a) Register

   b) Stack
   c) Heap
7. Übungsaufgabenteil (Teil 2)
8. Crash Kurs Assembler (x86 Assembler)
9. Crash Kurs C
10. Gegenüberstellung Assembler und C
    a) Ohne GCC Optimierungen
    b) 1 : 1 Vergleich Assembler C
    c) Mit GCC Optimierungen
11. Übungsaufgaben (Teil 3)
12. Tools Übersicht
13. Reverse Engineering Tools
14. Übungsaufgaben (Teil 4)
15. Patching
16. Übungsaufgaben (Teil 5)
17. Ghidra Interna
    a) P Code
    b) Sleigh
    c) Skripting (Python)
18. Übungsaufgaben (Teil 6)

Diese Struktur macht die Abwechslung zwischen theoretischen und praktischen Elementen erkennbar. Das verhilft auf der einen Seite zu einem angenehmeren Lernprozess und auf der anderen Seite zur einer abwechslungsreicheren Ausarbeitung des Kurses.

4 Vorgehensweise der Ausarbeitung

Zur Vorbereitung erfolgte die ausführliche Auseinandersetzung mit dem Buch namens 'Reversing - Secrets of Reverse Engineering' von 'Eldad Eilam' [1]. Dieses Buch wurde von Herrn Prof. Dr. Rer. Nat. Tobias Heer empfohlen. Dieses gibt eine sehr gute Einführung in den Bereich des Reverse Engineering. Es sammelt Informationen zu den Reverse Engineering 'Basics', weist aber ebenfalls in den Bereich des 'Applied Reversing' ein, welches spannende Kapitel über die Themen 'Beyond the Documentation', 'Deciphering File Formats', 'Auditing Program Binaries' und dem Bereich der Malware enthält. Zu dem enthält das Buch weitere Informationen über den Bereich des 'Crackings', welches ebenfalls innerhalb des Kurses behandelt wurde. [1]

Nachdem das Buch durchgearbeitet und somit die Grundlagen des Reverse Engineering erlernt wurden, konnte sich anschließend mit dem Tool 'Ghidra' beschäftigt werden. Dieses Tool ist mittlerweile Open Source und wurde nach jahrelanger Entwicklung von der National Security Agency herausgegeben. Um sich die Grundlagen des Tools 'Ghidra' beizubringen, wurde ein Kurs eines YouTube Kanals namens 'HACKADAYU' herangezogen. Dieser Kurs bündelte Theorie, als auch praktische Aufgaben. Somit konnten die Grundlagen des Tools 'Ghidra' erarbeitet werden.

Mit Hilfe dieser zwei Informationsquellen konnte mit der Ausarbeitung des Kurses begonnen werden.

Hierfür wurde ein weiteres Dokument angelegt, in dem die Gedanken für die Ausarbeitung des Kurses gesammelt wurden. Dieses kann ebenfalls innerhalb der Kursunterlagen gefunden werden.

Hierzu wurde erstmals die generelle Struktur des Kurses mit Herrn Prof. Dr. Rer. Nat. Tobias Heer abgesprochen. Herr Heer gab den Tipp, den Kurs iterativ aufzubauen. Aus dieser Idee heraus entstand die Einteilung des Kurses in mehrere theoretische und praktische Kapitel. Hiermit wurde ein Kurs geschaffen, der erweiterbar und anpassbar ist.

5 Kapitelübersicht

1. **Was ist Reverse Engineering?**:
   - Grundlegende Erklärung vom Reverse Engineering
   - Einführung in das Reverse Engineering
2. **Überblick**:
   - Auffassung von Anwendungsmöglichkeiten und verschiedenen Techniken des Reverse Engineering.
   - Überblick über verschiedene Themen des Reverse Engineering
3. **Rechtliches**:
   - Rechtliche Informationen bezüglich des Reverse Engineering.
   - Nicht jedes Programm darf ohne weiteres dem Reverse Engineering unterzogen werden.
4. **Übungsaufgaben (Teil 1)**:
   - Installation und Einführung in Ghidra.
   - Dazu drei einführende Übungsaufgaben, um das Reverse Engineering praktisch mit Kapitelübersicht Hilfe von Ghidra anzuwenden. Die erste Aufgabe bietet als Einführung eine Schritt für Schritt Anleitung.
5. **Vom Code zur Binärdatei**:
   - Erklärung des Prozesses, wie eine Binärdatei entsteht.
   - Hier werden die Informationen offengelegt, wieso und wie Informationen des Codes im Umwandlungsprozess verloren gehen.
   - Reverse Engineering ist der umgekehrte Mechanismus
6. **Systemkomponenten**:
   - Erklärung verschiedener Systemkomponenten und dessen Zusammenspiel mit Programmcode.
   - Das Verständnis ist wichtig, um die Grundlagen des Reverse Engineering zu verstehen.
7. **Übungsaufgabenteil (Teil 2)**
   - Erweiterte Einführung in Ghidra
   - Zwei weitere Aufgaben im Bereich des Reverse Engineering

- Die zu untersuchenden Programme sind selbst verfasst, um den Schwierigkeitsgrad besser zu bestimmen.

8. **Crash Kurs Assembler (x86 Assembler)**
   - Crash Kurs im Bereich von x86 Assembler.
   - Das Verständnis von x86 Assembler ist für das Reverse Engineering auf aktuellen Systemen essenziell.
   - Dies ist die tiefste Ebene der Repräsentation von Befehlen.
   - Viele Beispiele zum Verständnis der verschiedenen Konzepte von x86 Assembler.
9. **Crash Kurs C**
   - Crash Kurs im Bereich von C.
   - Ghidra besitzt einen Decompiler, der aus bestehendem Assembler Code eine C ähnliche Repräsentation generiert.
   - Einführung in Grundlegende Konzepte und Datentypen in C.
   - Ebenfalls eine Erklärung von C mit Hilfe vieler Beispiele.
10. **Gegenüberstellung Assembler und C**
    - Einige Gegenüberstellungen von Assembler und C anhand von Beispielen.
    - Praktische Vertiefung von Gelerntem aus vorherigen Kapiteln.
    - Im Reverse Engineering ist es wichtig, diese Gegenüberstellung zu machen und diese zu verstehen.
    - Dies bietet eine gute Grundlage im Verständnis von Ghidra und anderen Reverse Engineering Tools mit Decompiler.
11. **Übungsaufgaben (Teil 3)**
    - Drei Übungsaufgaben, um das Wissen weiter zu vertiefen
    - Zwei Übungsaufgaben im Bereich der selbstständigen Konvertierung von Assembler Code in C Code und andersherum (von Hand).
    - Ein weiteres Crackme von einer Website.
12. **Tools Übersicht**
    - Vorstellung einiger Linux Kommandozeilen Tools, die beim Reverse Engineering von Interesse sein könnten.
    - Ziel ist das Schaffen einer generellen Übersicht über vorhandene Tools, dessen Verwendung und Anwendung.
13. **Reverse Engineering Tools**
    - Vorstellung verschiedener Reverse Engineering Tools.

- Damit wird ein Überblick der verschiedenen populären Reverse Engineering Tools geschaffen.
- Zudem werden die Eigenschaften der verschiedenen Programme kurz erläutert.

14. **Übungsaufgaben (Teil 4)**
    - Eine größere Reverse Engineering Aufgabe.
    - Es handelt sich um einen TCP Server, mit dem mit Hilfe einer erfolgreichen Autorisierung verschiedene Dinge gemacht werden können.
    - Beispielsweise können im autorisierten Bereich Befehle auf der Seite des Servers ausgeführt oder eine Remote Shell gestartet werden.
    - Dieses soll als (immer noch als verhältnismäßig kleines) 'reales' Beispiel dienen, wie eine solche Reverse Engineering Aufgabe aussehen könnte.
    - Ziel ist es mit Ghidra eine der insgesamt zwei eingebauten Schwachstellen zu finden und diese auszunutzen.
15. **Patching**
    - Einführung in den Bereich des Patching
    - Durchführung und Erklärung eines praktischen Beispiels mit dem Kommandozeilentool 'VIM'. Hierbei sieht man, wie das Patching von Hand mit Hilfe zweier Linux Kommandozeilentools vollführt wird. Es wird dadurch erkenntlich, welche Aufgaben Reverse Engineering Tools wie 'Ghidra' übernehmen.
    - Ebenfalls Durchführung und Erklärung eines erweiterten praktischen Beispiels mit Hilfe von 'Ghidra'.
    - Beide Beispiele sind Schritt für Schritt Anleitungen zum Patchen der Binärdatei.
16. **Übungsaufgaben (Teil 5)**
    - Weitere Vertiefung des Kapitels 'Patching' durch zwei praktische Aufgaben.
    - Ziel ist das Verstehen und Umgehen von Sicherheitsmechanismen in zwei verschiedenen Crackmes.
17. **Ghidra Interna**
    - Einführung in Ghidra Interna, um grundlegende Konzepte von Ghidra zu verstehen.
    - Hierzu gehören Prozessorspezifikationen und Registertransfersprachen.
    - Es können neue Prozessorspezifikationen hinzugefügt werden, falls benötigt.
    - Außerdem kann mit Ghidra mit Hilfe von in 'Java' und 'Python' verfassten Skripts kontrolliert werden.
    - Anbei sind praktische Beispiele zweier Skripts, die in der Programmiersprache 'Python' verfasst wurden.
18. **Übungsaufgaben (Teil 6)**

- Untersuchung der populären Malware 'WannaCry'.
- Das Reverse Engineering spielt im Bereich der Defensive gegen Schadsoftware eine sehr wichtige Rolle.
- Anhand des Kurses wird das Wissen erlangt, Schadsoftware zu untersuchen.
- Hier soll anhand einer real existierenden Schadsoftware gezeigt werden, warum das Reverse Engineering eine wichtige Rolle im Bereich der Schadsoftwareuntersuchung spielt.
- Außerdem wird offengelegt, wie die tägliche Aufgabe eines IT-Sicherheitsspezialisten aussehen könnte.

6 Ergebnisse

Ergebnisse der Studienarbeit können unter dem zugehörigen Git Repository im GitLab der Hochschule Esslingen eingesehen werden.

Gegenstand der Studienarbeit ist:

- Skript des Kurses: Dies ist der Hauptbestandteil der Studienarbeit. Innerhalb des Skripts finden sich alle Kapitel, die in dem Kapitel 'Kapitelübersicht' beschrieben wurden.
- Übungsaufgaben und Source Code: Die Übungsaufgaben befinden sich ebenfalls im Git Repository. Anbei befinden sich zugehörige PDF Dateien, die die jeweilige Übungsaufgabe beschreiben.
- Folien: Es sind zusätzlich erstellte Folien für die jeweiligen Kapitel vorhanden. Diese Folien spiegeln den Inhalt des Skriptes wider.

7 Schluss

Alles in allem war die Ausarbeitung des Kurses eine bereichernde Erfahrung. Es bestand die Möglichkeit einer ausführlichen Beschäftigung mit der Thematik des Reverse Engineering, die zu einem großen Lernerfolg führte. Durch die Terminierung wöchentlicher Meetings gab es die Möglichkeit, Fragen über die Thematik des Reverse Engineering zu stellen und ebenfalls Informationen über eine erfolgreiche Ausarbeitung eines Kurses zu bekommen. Viele Unklarheiten konnten dadurch beseitigt werden.

Durch die Ausarbeitung der Übungsaufgaben wurde die Theorie praktisch vertieft. Somit bestand die Ausarbeitung sowohl aus theoretischen Teilen, als auch aus praktischen Teilen. Das führte zu einer ausgeglichenen Arbeitsweise und ebenfalls zu einer Verstärkung des erreichten Lerneffekts.

Das Ziel der Studienarbeit ist es, anderen Studierenden ebenfalls solch ein Lernziel zu ermöglichen. Durch den Aufbau des Kurses wird eine gute Lernkurve erreicht, da die gestellten Aufgaben immer komplexer werden. Da der Kurs generisch gestellt ist und wenige Vorkenntnisse für das Verstehen des Kurses nötig sind, kann dieser von nahezu allen Reverse Engineering Interessierten bearbeitet werden. Es soll so möglich sein, sich mit Hilfe dieses Kurses in die Thematik des Reverse Engineering einzuarbeiten und den erarbeiteten Lernerfolg in komplexeren Aufgaben einzusetzen.

Literaturverzeichnis

[1] Eldad Eilam, Reversing - Secrets of Reverse Engineering, John Wiley & Sons, 2011