The aim of the project was to learn reverse engineering. This involved getting a better understanding of x86 assembly code and learning both static and dynamic analysis. This would be accomplished by completing challenges from crackmes.one and programs written by others in the course. A report with walkthroughs of the challenges needed to be completed and a basic overview reverse engineering.

A better understanding of x86 assembly was achieved through research on the roles of registers and instructions. Both static and dynamic analysis were used in the completion of the challenges and are shown through the walkthroughs. Challenges ranging from level 1 to 3 were completed from crackmes.one. Additionally, programs written from others in the course were completed. A report containing walkthrough of all the challenges were completed and it also contains general concepts needed to perform reverse engineering.

A deeper understanding of x86 assembly was learnt through research and practical application. This included the responsibilities of registers and instructions. Static analysis was learnt along with a general understanding of dynamic analysis. Throughout this process, skills with patching programs were developed in controlling the flow of instructions. Skills with decryption were acquired by decrypting text from a Caesar and a modified Caesar cipher where each character is shifted a different amount. An important tool learnt was the appropriate use of gdb in performing dynamic analysis. Another tool learnt was binary ninja for viewing assembly code. Additionally, I had learnt how various ways of making programs more difficult to reverse engineer such as cryptography.

Overall, this was a great project to undertake due to the skills and knowledge learnt. Some knowledge learnt were useful in other courses, particularly COMP3231. Unfortunately, not all the goals set out, were completed as I was too ambitious and did not expect the learning curve required in the initial phases. The learning curve was in regards to understanding x86 assembly and the appropriate steps for reverse engineering. However, once that was overcome I was able to quickly progress from level 1, to 2 and then 3. The main goals not accomplished were completing level 4 and 5 challenges. Level 4 may have been achievable if more time were available. Advice for those that choose reverse engineering, is to familiarise themselves with assembly and begin with level 1 challenges with walkthroughs or tutorials.

A mistake made during this project was that dynamic analysis was learnt towards the end of the project. It is a very powerful method in reverse engineering and would have been useful at the beginning of the project. An issue faced during the project was that machine code executables from crackmes.one were not for MacOS but Linux. To circumvent this, binary ninja was used to view the assembly of the program on Mac and the program would be run on cse servers, where it was compatible, along with gdb.

All the evidence for the project can be found in the appendix below. The GitHub repository contains all the challenges accomplished along with walkthroughs and a report. [see Appendix A]. The file Reverse Engineering Report in the repository contains all walkthroughs and other information. The page on OpenLearning contains links for blog posts made in regards to my progress on the project along with other evidence. [see Appendix B].

Appendix A (GitHub Repository):
https://github.com/Nikil-Singh/Reverse-Engineering

Appendix B (OpenLearning Page):
https://www.openlearning.com/u/nikilsingh-q5u5yl/SomethingAwesomePage/