Aven Au z5208734

Something Awesome Poster

Summary:
The first half of the project is basically me creating a sorting program and a login program for my partner to solve and my partner will do the same. For the sorting program I must find out what the sorting algorithm was used while for the login program I must find the password to the program. The second half of the project is attempting harder reverse engineering problems from crackme.one. In my proposal I outlined that I must do two level 4 crackme.one problems for me to get a distinction and 1 level 5 crackme.one problem for High Distinction.

What I Achieved:

I created a bubble sort algorithm that does not exit early. For the login I use the password with all its ascii values increased by one to compare whether the password was correct. This would prevent people from using strings to just find the password and paste it in. I was able to find what algorithm that was used in the program by going through the decompiled assembly code. I was also able to find the password of the login system by going through the set of "cmp" that was comparing the input which the characters.
Solving a level 4 crackme.one problem was also one of the things I achieved. It was an obfuscation problem which means that a lot of the functions were meaningless and was there to set me off track.
However, I was only able to reach the half way point of distinction as I was only able to do 1 problem.
For each of the reverse engineering problems I have solved I wrote a detailed writeup that contains my thought processes and a walkthrough on how to solve the problems.
These can be found on my Github: https://github.com/Dollaking/SomethingAwesome.
I also posted my reflections and walkthroughs on my blog posts as well.

Update 1: https://www.openlearning.com/u/avenau-pnay4y/blog/SomethingAwesome1/
Update 2: https://www.openlearning.com/u/avenau-pnay4y/blog/SomethingAwesome2LoginSystemReport/
Update 3: https://www.openlearning.com/u/avenau-pnay4y/blog/SomethingAwesome3/
Update 4: https://www.openlearning.com/u/avenau-pnay4y/blog/SomethingAwesome4/
Update 5: https://www.openlearning.com/u/avenau-pnay4y/blog/SomethingAwesomeUpdate5/
Update 6: https://www.openlearning.com/u/avenau-pnay4y/blog/SomethingAwesomeUpdate6/

What I learnt & reflections:

This project made me more comfortable with assembly. Before the project, even though I had experience writing and translating C code to assembly, I was still very unfamiliar with it. Especially in the login and algorithm reverse engineering problems, I was forced to read each line of assembly code, attempting to understand it all. Even though I was not able to understand every single line of assembly I was still able to have a good grasp of the program that I was trying to reverse engineer. This also taught me, how to make it very hard for someone trying reverse engineer my programs in the future. An example is function names, I realise function names is a huge asset in reverse engineering as it makes life much easier on what the program is doing. If I make obscure/encoded function names than it would make life for the reverse engineer a lot harder.
For the level 4 crackmes.one I chose to do Obfuscation1 at random. This challenge was a huge time sink for me because I just threw myself in to the deep end. Coming from trying to read every single line of assembly code, it was a challenge because this problem puts out a lot of random code to throw me off. There was even a warning saying that the program was too big to view. This challenge taught me to look for things that are useful and that reverse engineering doesn't always mean reading through all assembly code. I can still understand the program by just looking at a few lines of important code. In the case of Obfuscation1 I must only go through the cmp commands because upon checking passwords, comparing must be done.

As I am writing this I feel like I didn't do a lot of problems, I was more in depth in one problem but I think I should have done more easier problems as it will give me more tools to work on harder problems. This lack of breadth also means I wasn't be able to be challenged in different areas. I realised that some reverse engineering problems require patching which I wasn't even exposed to. I was only exposed in trying to find the password/code that was in the program. If I was to do this again, I would prioritise on easier and unique problems rather putting my self in the deep end trying to solve a hard problem.