Hi Sam

More comments;

1. In general. You need to make references to books or resources on the web to show where you have learnt and researched the information you are giving. Provide lots more referencing please Sam.
2. I think anything you mention in the success criteria should be discussed and explained in the analysis and background beforehand. For example,
3. - ports, make a really clear section where you explain them
4. - arguments, show in your background what these are and how they relate.
5. - TCP ports, what are they used for, how do we connect to them
6. - parse either a coma ......explain parsing and why this is required
7. - CIDR, explain in more detail above.
8. In fact, success criteria 12 to 16 need explaining within the background to the problem, the list in 1.3 is just a numbered summary of everything you've researched and explained in the background. All the points in the success criteria need identifying and explaining more thoroughly, and then the list in 1.3 makes sense of what it is your programs are going to do. — 9-15: fair enough CIDR is a weird concept at any level so I'll go into more detail on what it is as well as parsing arguments, more detail on all the success criteria especially why they are desirable features etc.
9. In 1.4 you need to clarify where nmap sits in a PC software stack. And explain what you mean by scanning types. Is there a diagram and more publicly available details on how nmap works that you could copy and refer to? — I'm not sure what you mean by sits in the software stack and there is a book on nmap that is freely and publicly available on the internet as a collection of webpages. — Is nmap an application at level 7, taking user commands, but translating and running code at layer 4??? Whatever you can add to describe nmap in more detail will be great.
10. Data dictionary. You now start to think about how your code will work. A data dictionary is the first stab at what data you will store temporarily in Ram or what your application may commit to secondary storage. In the real world this is important to know so that the useable Ram would not be compromised. In this section it normally uses a table where you identify the main data types and the amount of data for each scenario. If you were on a rocket going to mars the Ram is limited!! Every app we write has been carefully vetted to ensure we don't run out of space!! Joke!! — I believe my scanner uses very little memory but I could go into some of the specific data structures I have used and how expensive they

are? — Exactly. You must be using some registers and data storage for all the data you receive etc.

11. 1.9 - you've made too many references to why you have done, this section is supposed to be how you are going to do it. Firstly, make mention to the prototyping you did. This is where I asked you guys to take lots of screenshots of when you were trialling things. You need to re-write this to imagine you are writing down how you propose to approach the design before actually going ahead with it. — ok so I actually still have a lot of the screen shots so I think rewriting this should be ok, I can show each of my prototyping steps for each one etc.

12. Then in the data flow diagrams you can identify what data packets are created in which modules of your code and how they are sent to the lower layers of the OSI model via the python interfaces.

I hope this helps Sam. I know there is a lot above. Read through my ideas a few times before you start so that you know where to add things and I hope you agree.

Keep going!! It's excellent. I just want the examiner to be really clear on the amount of excellent knowledge and research you have gathered