1 notes

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.~~

13. ~~I love the ladder diagram on page 7 but can you show a very simple version with just the main packets within the explanation on page 5 within the paragraphs at the top of the page. It will really help the wonderful explanation.~~

2 section notes

- 1.2 What does it mean for a firewall to drop a packet. Explain what tcp rst packet and what this doesn't help. Some very simple ladder diagrams will be huge benefit for your explanation of the sequence of messages on pages 9, 10 and 11 These are great explanations, are there any documents you can refer to such as the protocols!!!! V important to get these references to back up your wonderful research. If you only have them on paper we can get them scanned or just use screen grabs. I love your description on page 14!!! Make the title of 1.3 just 'Success Criteria'

- 1.4 Could you get some screen grabs of what nmap looks like. Are there any others? Doesn't wireshark also do this Sam?? Get some reviews of nmap, show some blurb from 'its' website?? What will yours do differently?? If not anything that's fine, but it would be good for you to explain what you are doing differently by you actually writing the code to generate the platform on which to perform the testing.

- 1.5 Did you get chance to speak to mr Blake or mccarroll?? They could be mentioned here. But it's fine except I wondered if you could explain a bit more here what you mean by a banner.

- 1.6 Add what data your program will need to create and store during a straightforward running of one or more of the signalling sequences you are intending to run. This is a very straightforward section that students often get confused about. It's to highlight what data and the amounts of data we need to manage. The recent Boeing disasters are the result of software malfunctions and may well be due to storage requirements of something quite simple. You don't need to go into massive detail in the analysis. But it's important to show you have though through the data storage requirements.

- 1.7 The data you show in 1.6 needs to be created somehow and some programs may pass it around several clients or databases or software modules. What will you programs do with the data, are you going to store it? If part of your success criteria was to hold all of the data from a signal in a file, so it could be compared against success or unsuccessful packets in the future you would explain here where it is being saved and draw a simple filing structure. Are you storing any long term data??
- 1.8 Where does the data originate such as IP addresses. Or ports. Are they entered by the user, or generated by the program or retrieved from a file? How about arguments? These are all entered by the user I believe. But explain here. It can all go in a simple table.
- 1.9 You need a diagram to show how you your code will link with the other modules. You could also show a simple example of your python here. This section is a general overview of your solution and what you will write and how you will test it. Go bonkers here on the level of detail. I would like to see a lot more, think about how you would explain what you've got to do to someone like Mr Tuson. It's supposed to have enough detail so a new person picking her analysis section up could go ahead with the design section (but they would know how to program - not Mr Tuson)
- 2.1 More diagrams of how your modules will connect. With ladder diagrams. Lots of ladder diagrams And explanations of the processing of the data when you receive the responses/acknowledgment or don't receive correct signals??!?
- 2.2 Screen grabs of examples from your prototypes Show screengrabs of simulations or nmap to show how they do it??
- 2.3 Flowcharts of your code v important. As many as possible!!!!
- 2.4 Yo must have to validate the arguments and the commands????? Explain in detail how this will work.
- 2.5 Excellent! Are there more to add????
- 2.6 Details about the data storage in run time and persistent, as for analysis stage but here you will know 'exactly' what you will need
- 1.1 a simple diagram or 2 on page 4 and/or 5 to show the flow of signals and the laptop/server just to give an overview. I still do think there is opportunity to add more ladder diagrams to aid the excellent description
- 1.7 this shows where does the data used within your program originate and where is it stored, it should be a quite simple diagram for you.

- 1.10 you need more description on how you will actually test it, maybe an example of a command and arguments, and what this will run on i.e one laptop, or a whole network of supercomputers, you need to basically identify the test environment, any ip addresses you would use or need setting up. Give an overview of how it is set up in a diagram?? What software would you need to run it, what applications on the laptop, what commands etc

- 2.1 as mentioned earlier today, the nice description in 2.1 contains information that is sort of sprung out of the blue, this needs mentioneding in the analysis and potential solutions, apologies of you have and I have missed it.

- 2.2 I'm a bit confused, here is a list what netscan supports, are you using the same commands and arguments? Show here the commands your code will run, but either here or in the user interface section or in 2.4 you need to list all of the commands and the parameters the user can enter, and how to get it running, and the error messages that will arise if the user enters an illegal argument or command

- 2.3 add flowcharts of how your code performs its internal processes, I'm not sure how we link in to the ladder diagrams, i.e. you code flowhart initiates a signal, it awaits a response, it then acts on the type of signal and data accordingly, but it would be nice to show the link between your code and the signals being sent and received. Have a think.

- 2.4 please do make a full attempt at identifying every command and argument that you want to be able to process, clearly explain the system to ensure the user can't add incorrect arguments and how this is handled,