

Luke Demi

1. How many prompts did you need to use complete this assignment?
 - a. I utilized a total of 20 different prompts to complete this assignment. These prompts were used to address various aspects of the implementation, including error handling, socket communication, command parsing, and hash generation.
2. Provide a summary of your approach to splitting the 'echo service' problem up into different prompts.
 - a. I at first started off by simply trying to write a server and client that could communicate with each other through port 1337, which was relatively simple given that I have done it before. After that was all finished up, I started to layout the code commands in the server like echo, repeat, length, and hashing. I added the echo and length functionality first because those were relatively easy, and then figured out the hashing functions. I also updated my client so it could take in multiple server response threads for the repeat command, which I then got working as expected. I also had AI develop error handling for all of this code and implemented. Once I finished this up, I removed testing code and made other quality of life changes to make the program run smoother. By splitting up this complex assignment into different parts, I was able to tackle the issues one at a time which made it easier to code and develop.
3. In your estimation, how much of code generated by the AI/LLM worked as expected?
 - a. I would say about 40% of the code that I used generated from the AI worked quite well. The AI particularly found a lot of use when trying to figure out how to do the different types of hashing, which basically taught me how to understand wincrypt commands and generated code based on the context I gave it. This code ended up being a part of the final version. Also, I used AI to develop error handling for much of my program, since that it is actually good at. I think that 100% of my error handling in this program is AI generated. A large share of the code developed by the AI however did not work, and caused a massive amount of compilation and runtime errors.
4. In your estimation, how much of the code generated by the AI/LLM ended up in the final version of your code?
 - a. I would say about 25% of the code generated by the AI ended up in the final version of the code, particularly in regards to error handling and the hashing functions. Those were the sections I did not particularly know much about so I relied heavily on AI to develop those areas for me. I did not implement all of the working AI code into my final project though, because I found in many cases I could write code that was far more efficient and readable than the AI did.
5. In your estimation, what percentage of the code was written (or adapted/paraphrased) by you and what percentage was directly taken from the AI/LLM output?

- a. I would estimate that about 30% of my code was written directly by AI and the other 70% of the code was written by myself. A large portion of the code made by AI was specifically for terms of error handling, while a lot of the program functionality I wrote myself (except for hashing).
6. Were there any sections of code generated by the AI/LLM that you did not understand? If, indicate what those code segments were.
 - a. Yes, when I first asked the AI how to develop an md5, sha1, and sha256 hash and it used wincrypt to achieve this task. I did not understand what any of the wincrypt functions did or meant at first, but learned them as I started to use them. I also did not at all understand how a lot of my error handling worked, but they worked effectively so I never really learned more.
 - b. More specifically, these lines of code for hashing:
 - i. `CryptAcquireContext(&hProv, NULL, MS_ENH_RSA_AES_PROV, PROV_RSA_AES, 0);`
 - ii. `CryptGetHashParam(hHash, HP_HASHVAL, hashBuffer, &dwHashLen, 0);`
7. If the answer to #6 was yes, what do those code segments do?
 - a. The `CryptAcquireContext` code acquires a handle to a particular key container within a cryptographic service provider and the `CryptGetHashParam` code retrieves data that governs the operations of a hash object.
8. What benefits, if any, did you find to using the AI/LLM you selected in solving this problem?
 - a. Without AI here, I would have been extremely hard to figure out why my program was getting different errors that were unclearly explained in visual studio. There were multiple times that I received a specific error code, which I then asked the AI about and had it explained to me. I was able to understand and resolve my errors because of this tool. Mainly, it sped up the process.
9. What pain points, if any, did you experience when using the AI/LLM you selected in solving this problem?
 - a. The AI sometimes was unsure what I was asking, and sometimes did things I exactly told it not to do. There were multiple times I told the AI to develop this portion of code without doing this specific thing, and then it went ahead and did it anyways. This is why a lot of my code I ended up writing myself, because of overall AI incompetence.
10. To the best of your knowledge, is your code vulnerable to any buffer overread or buffer overflow vulnerabilities? Back your answer up with specific references to your code. Use specific line numbers if appropriate.
 - a. My code is vulnerable to buffer overflow vulnerabilities because of some of the limitations given for this assignment. In particular, we were told to not use anything besides `scanf`, even `know fgets` is much better for mitigating buffer

overflow vulnerabilities. Also, the receive functions could also be vulnerable to buffer overflow issues, since the `srvResponse` buffer can only take in 1024 characters. If the user types in more than 1024 characters, there will be memory corruption.

- i. `scanf("%[^\n]", userInput);`

- ii. `bytesRead = recv(clientSocket, srvResponse, 1024, 0);`

11. Are there any other comments you have or insights you gained about using an AI/LLM to assist in coding from this assignment?

- a. I've really learned for this assignment that AI is not really helpful for general coding, it's only really helpful if you don't understand how to achieve a portion of an assignment. Whenever I had the AI generate code I already kind of knew how to write myself, it sucked. I only tried to rely on AI if I genuinely had no idea how to do something, like the hashing and error handling.