*Menno Brandt*

*As a peer reviewing student, you will only see comments written by you.*

First of all, I must express my respect to everyone in the group; this is undoubtedly a highly accomplished piece of work. In fact, I was quite impressed when I ran code of this scale within the scope of a group assignment. Out of all the group project code files reviewed by me and my team members, only yours has achieved such a high level of completeness. Just the server file alone contains more than 900 lines of code.  
  
As for the vulnerabilities, aside from the RSA key vulnerability, most of the other issues can be resolved through modifications. For example, the lack of sufficient validation of client and server connection messages can be addressed by adding integrity checks to the data fields of both. Additionally, the security of the file upload feature can be improved by restricting upload paths or enforcing content safety checks. The vulnerability in the maintenance mode command can be fixed by introducing authentication mechanisms in the handle\_public\_chat function. There are also some vulnerabilities in the client file, such as the WebSocket connection not being SSL-protected and the same RSA key issue as in the server file. These vulnerabilities are all well-documented and can be traced effectively. Overall, great job!

Isaac Joynes

*As a peer reviewing student, you will only see comments written by you.*

How should I put it… It’s an excellent piece of work, really well done, with a high level of completion. Regarding the vulnerabilities, aside from the RSA key vulnerabilities that I’m not familiar with and the WebSocket connection vulnerabilities present in both main codes, there is also a lack of file type and content validation in file uploads and downloads. This can even be exploited in file upload and download mechanisms using curl commands for injection attacks. It’s fair to say that most of the vulnerabilities arise from missing validation mechanisms in the code, making it susceptible to direct attacks. Also, about the README file—I remember it was supposed to be in .md format, so why is it in .txt? I suggest changing that accordingly.

Samuel Hunter

*As a peer reviewing student, you will only see comments written by you.*

It's an excellent piece of work, very well done, with both content completeness and vulnerability aspects being commendable. As for the vulnerabilities, aside from the RSA key vulnerabilities that I'm not particularly proficient in, the most interesting one, I think, would be the fully exposed debug information vulnerability. This one is quite fascinating—I essentially used this vulnerability for the entire vulnerability detection process afterwards, so I suggest addressing this issue. Additionally, there are various unchecked input handling vulnerabilities, along with the identity and message transmission vulnerabilities that share similar issues. These are all due to the lack of validation mechanisms. If you want to fix them, you can add corresponding validation steps. Moreover, this code doesn’t use SSL encryption, which fits the requirements well. Overall, apart from the unexpected debug vulnerability, the other vulnerabilities have been implemented in a manner that meets the specified requirements.