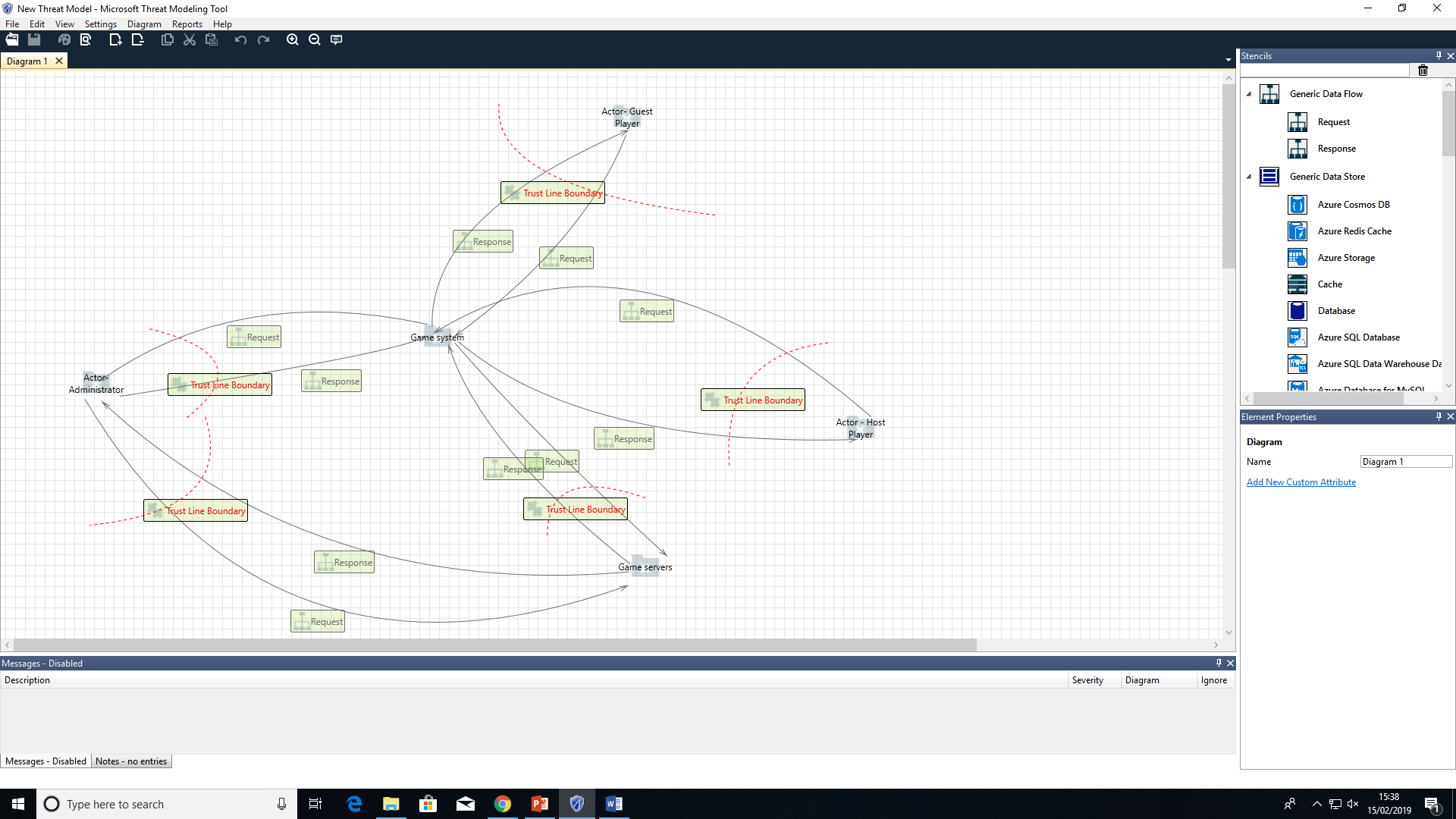
Threat modelling report:

Group 17

System overview diagram:



STRIDE analysis:

Spoofing:

There are a number of possible problem areas with the system when it comes to spoofing.

The main one that would affect the end user would be people using loopholes or errors in the workings of the system to gain access to a user’s social media. Due to the game allowing you to post on a player’s social media from inside the program, we would need to have a high level of user validation before allowing a user to post anything. A further level of this would be requiring a log in every time a user wanted to post even if they were previously logged in on another session.

Another possible spoofing problem, this one affecting the game system itself would be a player somehow getting access to the game server by spoofing a system admins login information. To combat this any admins would need a further level of authentication and validation than any users.

Authentication and validation would need to be taken seriously on all fronts due to the possible exchange of money in the app itself. With players being able to gain access to an in app marketplace, the login information and identity of each player would need multiple checks before being allowed purchasing access.

Tampering:

There would be a real possibility for tampering in the game system due to the back and forth transfers of data packets to the game servers themselves. These take the form of updates for the game and game packs purchased by the user but precautions would have to be taken to ensure that these were the only thing being transferred. For example if the user was able to send their own transfer request to the game servers instead of an update request they would be able to tamper with confidential files. A counter measure of this could be requiring a level of encryption on the files and data requests being sent. This encryption would only be known at the server end so anyone tampering would not be able to gain access to it. This could also be achieved through a sophisticated validation system for any data requests.

Repudiation:

The risk of repudiation in this system is seemingly low in that the user would do the majority of their usage of the system in one session. Nevertheless there is slight risk in that there is a risk of it for every system that has an online aspect. A user could make a move in the game and then deny it theoretically but there would be minimal reason to do this other than to cheat at the game. There would in theory be no security issues arising from this but nevertheless this could be combatted by having a log of player moves that can be seen at all times ensuring players had accountability for their actions.

Information disclosure:

There is a risk of information disclosure with this system in a number of ways. Most notably of that being the transfer sensitive data over networks. While not held in any form in the servers of the game itself, the game does act as a conduit for the user to access systems that require passwords such as the game store and the social media accounts integrated in the game. To reduce risk of this transfer of data being the weak point that is exploited in exposing data, the game could encrypt the sensitive data for the transfer period and decrypt once it has reached its destination. Another aspect of information disclosure to consider is that of the updates sent from the game servers. These would also have to be secure to limit risk of players learning information about the workings of the game that they are not supposed to be knowledgeable about.

Denial of service:

The servers of the game are at risk from a denial of service attacks due to the passing back and forth of data in the form of updates and score information. The user requests for these could be exploited by those that wanted to crash the servers or even the game itself. While this is an issue with the majority of online server based systems it still needs to be addressed for this particular system. A counter measure for this could be ensuring users cant request any data if they already have it. For example if they have the most up to date version of the game they wouldn't be able to update it any further. The validation for this would have to take place in the system itself rather than requiring an outside check.

Outside of that countermeasure the only other way to combat possible DOS issues would be to limit the amount of data transferred overall, by either limiting the size of the updates themselves or the frequency, only allowing users to request an update when one was available.

Elevation of privilege:

There is a possible risk for elevation of privilege to be exploited within this system. There are users that would have more access than others, e.g. system administrators and host users. This could be eliminated by restricting what these higher level users could actually do, for example the only extra ability of being a host would be creating the game itself, and then after this stage they would be essentially reduced down to normal player status. Aside from this the game would just have to have strict user account limits and policies, allowing the user to only have access to the bare minimum of data required to access and play the game and save any higher level functions for admins. Low level users would have very little file access privileges.