May 12, 2020

Improvements & Feedback

Due to the nature of the program, encryprtion of the data isn't that important, but preventing modification is. The only real improvements that would be necessary are in regards to the hashing and encryption/decrypiton of the information.

We need to replace the hashing function with a unreversable one, since I wouldn't be supprised if the one I used is weak. For the encryption, obviously something a lot stronger then an XOR. The biggest catch for this is that the signature hash needs to be reproducable between runs, such that the same info reproduces the same signature.

However I believe it's biggest strength is that an attacker, to actually make use of modifying the data, not only needs to know how to generate the signature, but also how to encrypt it properlly. Revealing the data is easier then modifying it, and at least for this program, isn't that big of an issue.

Even thoug the XOR function was found, it didn't have any "strings" that were easily identifiable, and the numerous calls inside made it look more complex then what it actually was, which discouraged me to whether or not this was correct. Finding it twice though definitly gave me more confidence though, which can be changed if we go to a better encryption/decryption algorithm.

Some other improvments that could be made is to scrap the "command" system, since finding the "exit" line was fairly easy. Essentially, hiding when the load and save occurred could help prevent the functions being found. The load was easy to find since it was at the beginning, and from dynamic analysis, we determined that it occurred before an big panda was created.

Finally, changing the actual contents of the file. Discovering the order of the information being saved matching the contents of the code allowed me to determine and confirm/increase confidence of the load, hash, and encryption order of the data, which let me deduce the functions being used. Somehow enterwining the information together or something else might at least make it less obvious.

As a side note, having easily readable strings might be an issue. It "might" be worth the time and effort to also encrypt all the "strings" inside (the hard coded ones such as "player.char") and using a decryption function on them when they are actually used. That way all strings will look like giberish when disassembled.