Team: Horus
Names: Calin Farmer, Nathan Granade, Cordus Bailey, Casey Lee, Josue Gaona, Brianna Butler, Braylen Simmons

Challenge 5:

What we did: We started by getting the epoch time from discord to use on our modified timelock program. We ran '**echo "2022 10 28 11 00 00" | python3 modifiedTimelock.py**' in the terminal to get the first two letters of the hash, last two digits of the hash, and middle character of the hash. We logged into the ftp server using the IP given in discord and the username and password from the pdf with the XXXXY replaced with the timelock output. Once in the ftp server, we moved to the 'FILES' directory and saw two files called 'k3y' and 'c1ph3r'. Using the commands passive, binary, prompt, and mget * we retrieved the two files from the ftp server and disconnected. We then used these two files on our xor.py program by changing keyFile to be read to 'k3y' in xor.py and giving it the file 'c1ph3r' to process. We directed this output into a file named '1', later renamed to lifeAlertBracelet, which gave us an obviously stegged file. By reading the pdf, we determined the offset of the first file had to be either 2 or 2048. We chose to use 2048 as the offset due to us using the byte method and the picture being messed up at the bottom and went by powers of 2 till we got the correct interval. Using the 1 file we ran our steg program by doing '**python3 steg.py -r -B -o2048 -i8 -w1 > 2**' to get the first file from the stegged image called '2' which was later renamed to 'cyber'. For the second file to get from the stegged image, using the pdf we started with an interval of 2 and went through powers of 2 plus 1 for the offset. We did this until '**python3 steg.py -r -b -o513 -i2 -w1 > 3**' gave us a readable output called '3' which was later renamed to 'constitution.txt". To get the secret message we used the output '3' on our steg program. We determined the interval to use by counting the letters in between each mistake in the text which gave us 64. We then determined the offset by figuring out where the first mistake was and trying a few values above and below it until we got 96 as our offset. Using the offset and interval on the command 'steg.py -r -B -o96 -i64 -w3 > 4' gave us the output file '4' that contained the secret message and was later renamed to 'secretMessage.txt'.

Secret Message: "Scientia Potentia Est!"

Casey: I tried to login to the FTP server until we were told that only one person can do it at a time. I also tried to understand what was meant by "counting is your friend" when dealing with the messed up declaration of independence text. I tried counting how many total mistakes there was to see if I could notice any pattern, but that did not work for me.

Cordus: I used the epoch on discord to get the output from the modifiedTimelock to get the current password for the ftp server. I logged into the ftp server and got the 'k3y' and 'c1ph3r' files to use on the xor program. I used xor program and retrieved files to get a stegged image. Using the steg program I retrieved the 3 file outputs, the cyber image, constitution text file, and the secret message file.

Braylen: I ftp’d into the server, had the correct username, but couldn’t get the password. So, I typed user and entered user as the username and also didn’t have the password for that user either. I got stuck. I noticed that there numbers on the side of the username after a failed attempt (531 and 330), yet I don’t know what to do with it unless it refers to permissions for the files.

Brianna:

Nathan: I attempted (and failed) to login to the FTP server using the timelock program. After getting stumped on that, I attempted to follow what Cordus was doing but fell behind pretty fast, successfully adding nothing to the team this challenge other than writing the timelock program.

Josue: I attempted to log into ftp server, but I failed. I then tried to steg all the photos and text files being put into our group chat. I struggled with finding the right offset and interval numbers.I then waited for the rest of the team to get the secret message and send it.

Calin: I started by attempting to manually brute force the original ftp server given by the early access document. After failing to do that I followed by attempting to use our team’s steg program to decipher the c1ph3r file with the k3y file and also using strings to see if I could find a hidden hash or phrase in the header of the file. I also tried reverse engineering the c1ph3r file with ghidra and failed at that as well and before I could attempt anything else our group had gotten the hidden phrase and messaged the instructor through our teams discord channel.