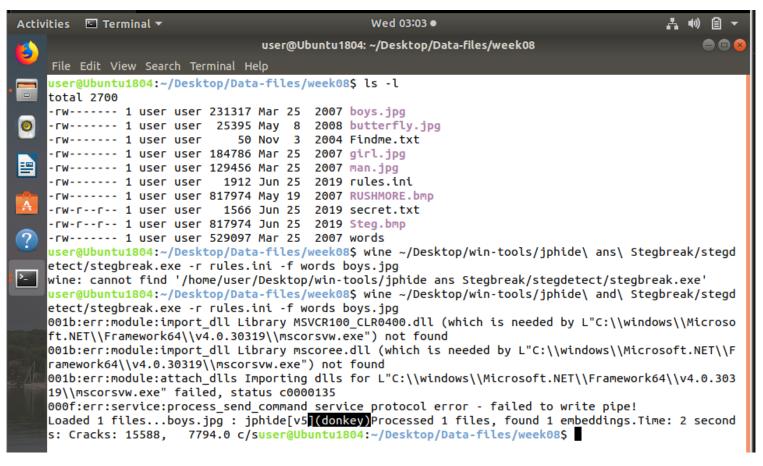
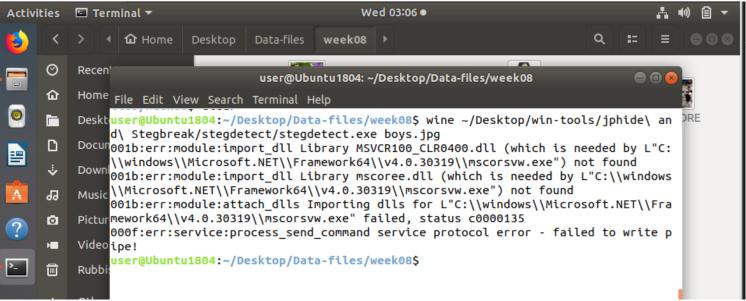
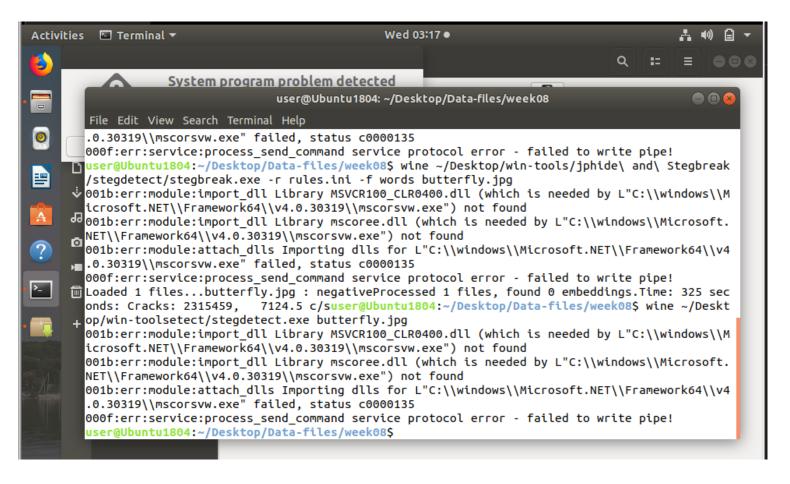
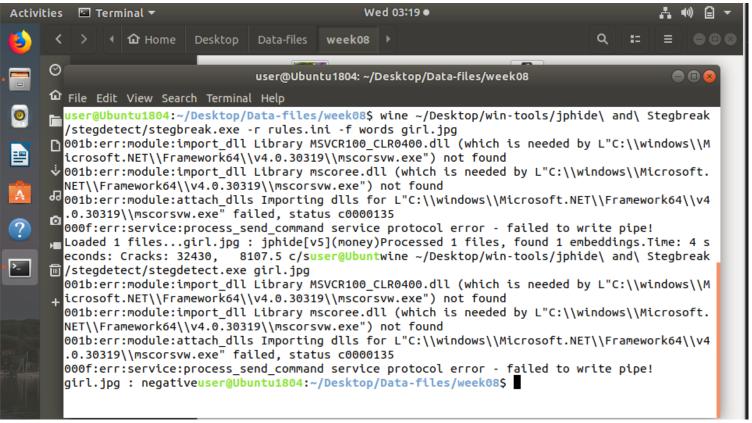
Introducing Stegbreak

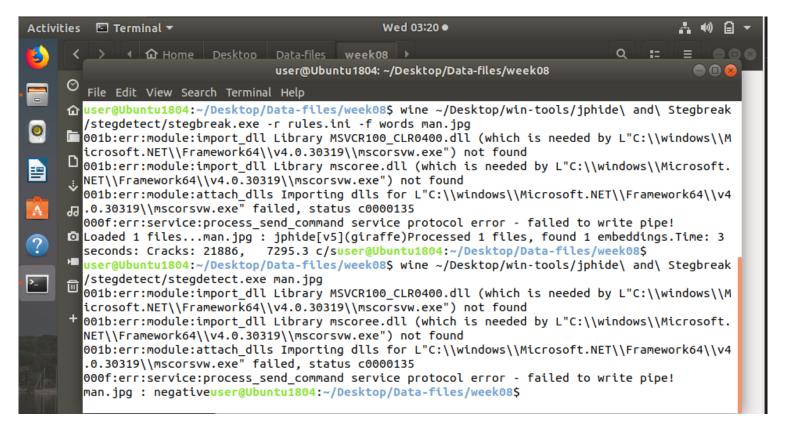
Initially, I was presented with Stegbreak, an automatic instrument for identifying steganographic elements within JPEG pictures. I performed dictionary attacks on images using the wine emulator and Stegbreak to uncover passwords that were hidden. I looked at the vocabulary file this program utilizes after getting to know it.

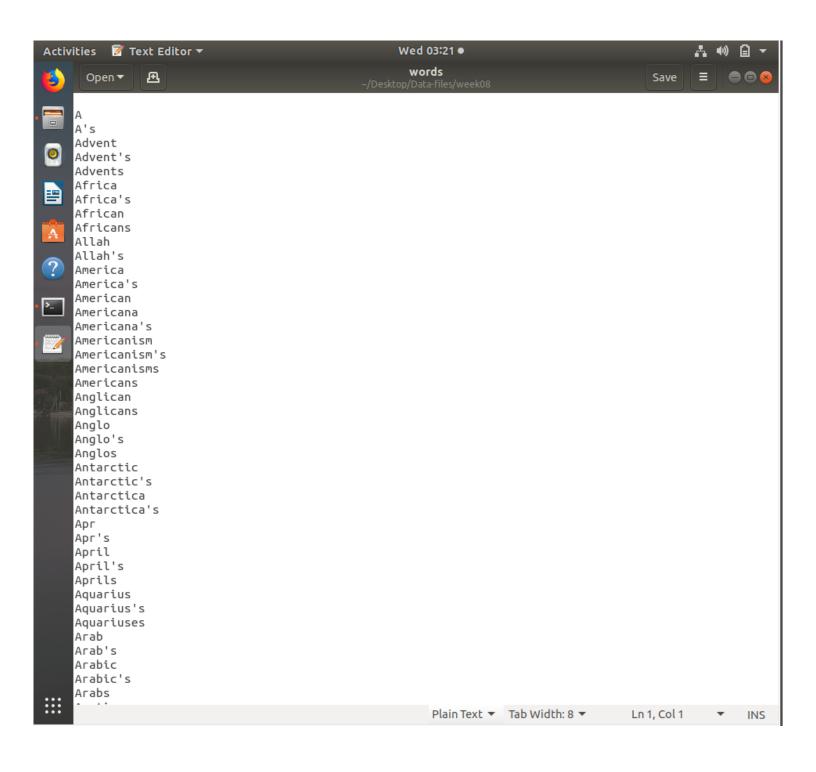








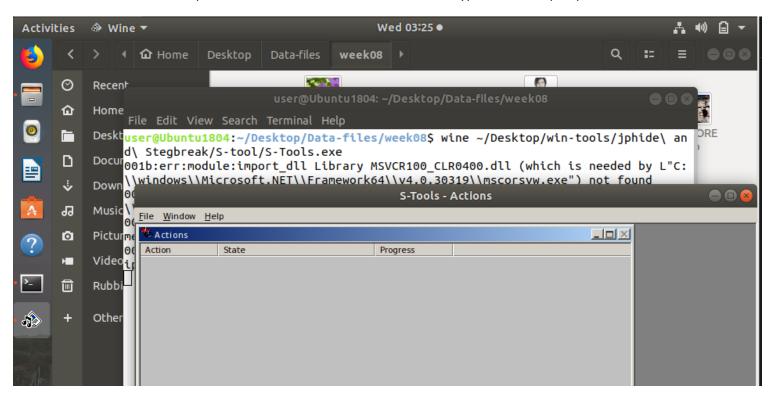




Introducing S-Tools

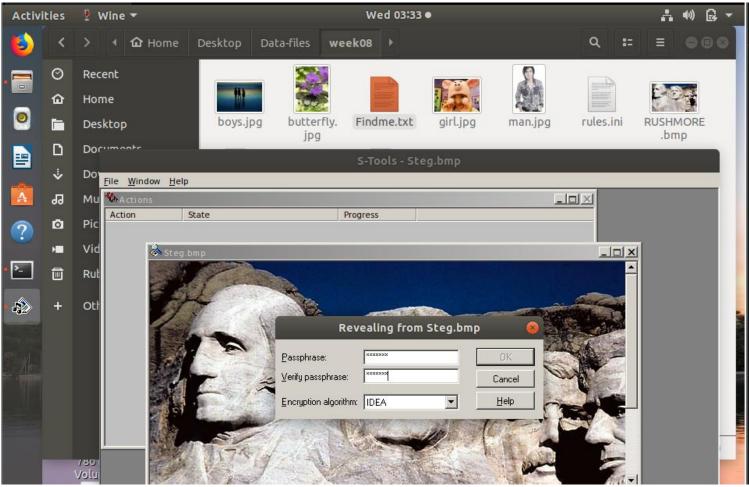
S-Tools, a GUI-based utility for embedding and exposing hidden content in image files, was the next program I looked at. I experimented with various encryption techniques to conceal and disclose text data inside of pictures.

In this I used the rushmore.bmp file and hide the findme.txt with IDEA encryption and the passphrase was FREEDOM.

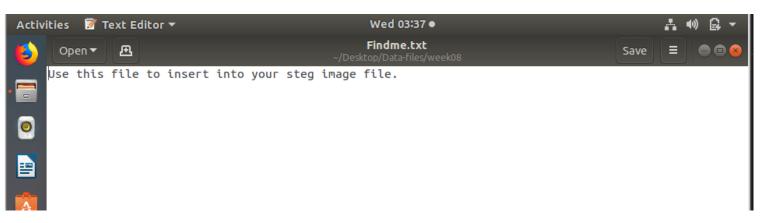




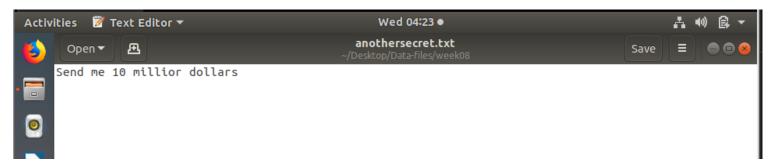




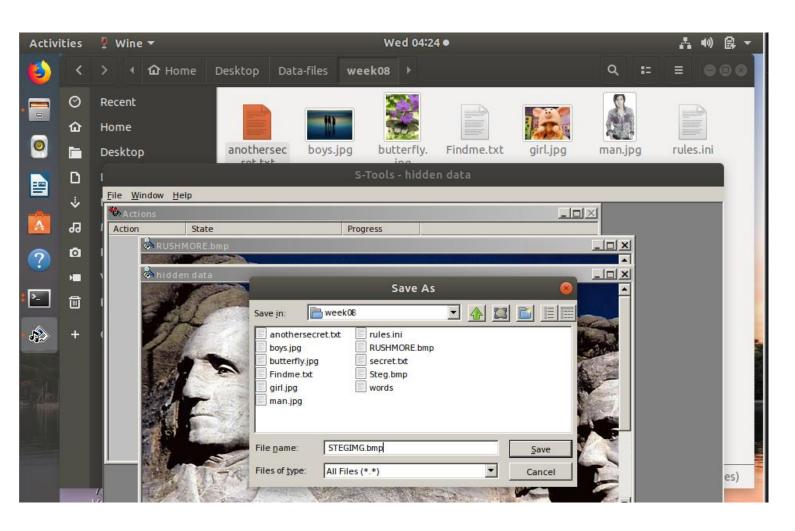




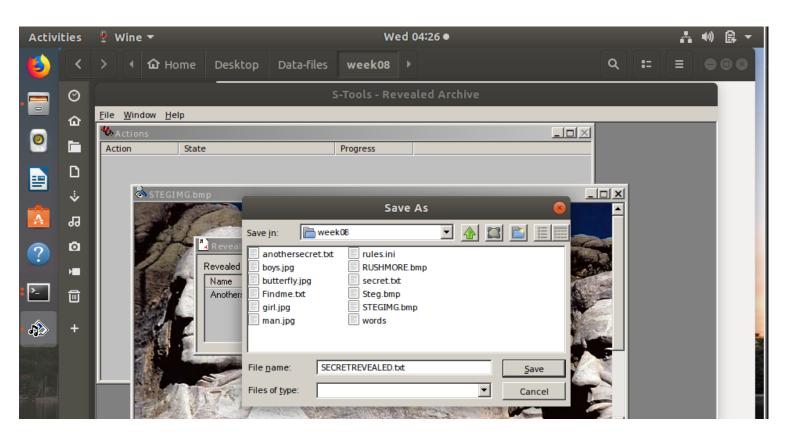
Then I created my own secret and hide it in the rushmore.bmp file. But this time I used the triple des encryption and a different password to hide it and then recovered it to see how it works.







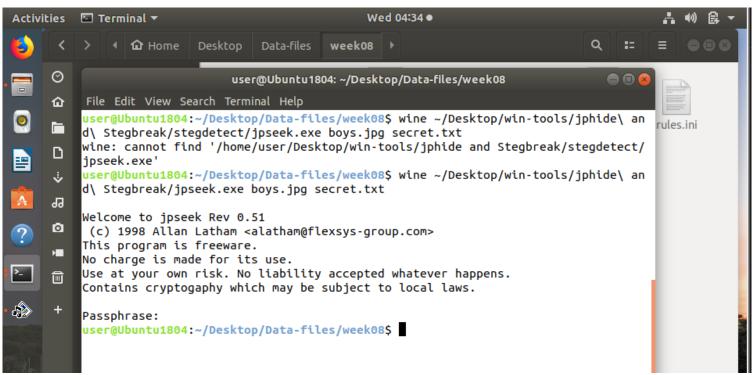


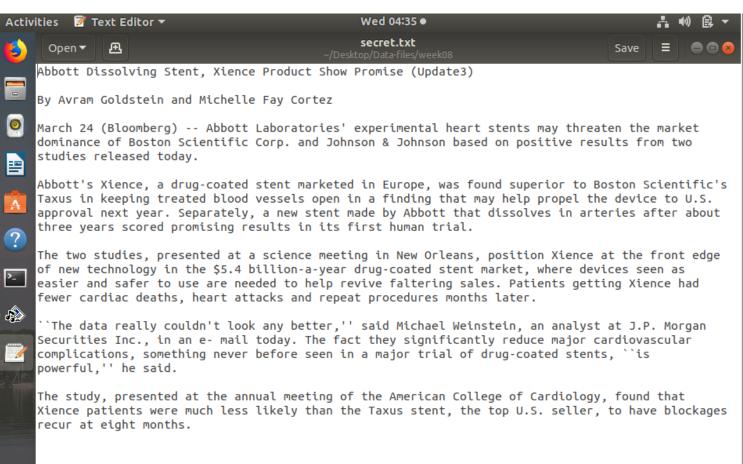




Using jpseek to Recover Steganography Images

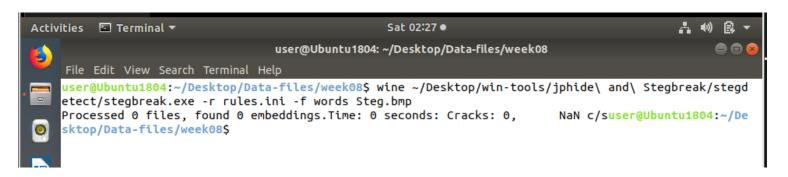
Finally, I used previously acquired passwords to extract concealed contents from photos using jpseek, another Windows utility.

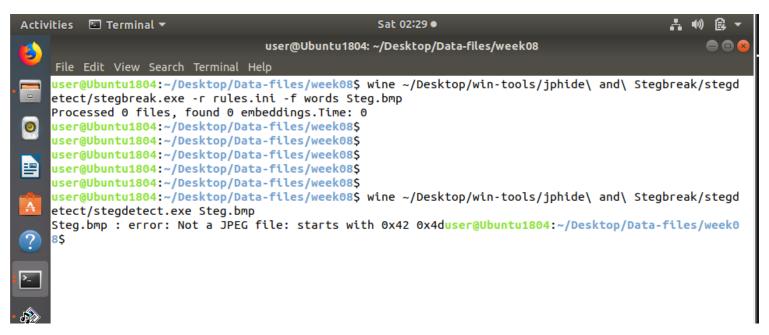




Forensic Tasks

My final task was to attempt to crack a picture I had processed using S-Tools using Stegbreak and to determine why it worked or didn't work.





The reason I didn't work is because it wasn't a JPEG file. The file I used was a .bmp file, so this tool stegbreak might only be used with JPEG images. The tool identifies what type of file it is with the header information of the image.