Steganography Detection Tools

Joshua Lopez

California Lutheran University

IT-540 Digital Forensics

Wayne Snyder

December 19, 2024

Steganography Detection Tools

Stenography is the art and science of writing hidden messages(Easttom, 2021, p. 120).

This paper evaluates three tools used for steganography detection. I chose S-Tools, DeepSound, and OpenPuff. My initial research began with searching on YouTube for video demos of these tools to get an overview of their use. We will describe each in further detail, listing its advantages and limitations. Lastly, we will make a recommendation with justification for our choice.

DeepSound’s layout seemed user-friendly. It allows for data to be hidden in sound files. After the third try, I could encode the data into the wave file. It's less intimidating than robust tools like FTK, E3, or Autopsy. The app's buttons are a nice size with a clear function description. The application has only two output windows, all contributing to a feeling of simplicity. A forensic examiner would easily understand how to collect carrier data, add secret files, encode, and extract secret files. It took me three attempts because I was not used to the flow of the app's procedure. After the second round, I moved through functions smoothly and logically(ZanoList, 204).

S-Tools app was not very user-friendly. There were no visible functions on the app or in the file menu. If it weren’t for the YouTube demo video (Kan et al., 2015). I would not know how to use it at first glance. The user must drag and drop the carrier file and the data into the application window. A window pops up where you can add a passphrase and the encryption algorithm. In the Hidden Data Window, right-click to save the file with hidden data to your machine. Now, drag and drop files onto the application window, right-click, and select Reveal. A window pops up where you can enter your passphrase and encryption algorithm. For this exercise, I chose Triple DES. Right-click on the reveal window and save the revealed hidden data file to your machine as a text file. Open the file on your machine to see the original concealed data.

I wasn’t too impressed with S-Tools. There was a lot of manual copying back and forth, which made the process seem tedious. I will say that I was pleasantly surprised to be able to choose the encryption algorithm. I bet the forensic examiners do as well. There must be a significance in having the functionality. Perhaps the algorithm is best chosen for a particular file type or the method's security. Either way, the more functionality/options allow for further granular/specific tests.

OpenPuff was a user-friendly multi-option app. Import your carrier file and data to be concealed. You are then given the option to insert 3 uncorrelated data passwords. Additional security of cryptography and scrambling. Import your carrier file data file to be concealed. You can select the bit type for the data. In this exercise, we chose a JPEG. Select Hide Data to conceal it within the JPEG file and close the window. Back on the first screen, select Un-hide. Enter your password, import the carrier file with the hidden data, select bit type, and then Un-hide. Visit your local folder where you choose the output. You will now see the data file revealed. Open it to confirm the original data was collected(Luse, 2014).

In my evaluation of these three tools, I learned a lot about data hidden in different types of files. My preferred choice would be OpenPuff for its user-friendly, intuitive application. My evaluation was done on a jpeg carrier. Other supported channels are images, audio, and videos. The data is encrypted, scrambled, whitened, and encoded before carrier injection. Up to 256MB of hidden data split among multiple carriers can be un-hidden (EmbeddedSW, n.d., Features section). Its ease of use would allow a beginner user like myself to grasp the procedure flow rather quickly. An experienced Forensic Examiner would breeze through hidden data analysis on this easy-to-use, self-explanatory tool.

References

bing.com. (n.d.). *Bing Images*. Retrieved December 17, 2024, from <https://www.bing.com/images/search?view=detailV2&ccid=Gp0GOk0O&id=1EB99D970DAC192EC4CE7BCEB9C0519A09CAEFF7&thid=OIP.Gp0GOk0OHZ6TEncJE8insgHaEo&mediaurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.1a9d063a4d0e1d9e9312770913c8a7b2%3frik%3d9%252b%252fKCZpRwLnOew%26riu%3dhttp%253a%252f%252fwww.superiorwallpapers.com%252fdownload%252fbasket-full-of-fresh-fruits-1920x1200.jpg%26ehk%3dCrfVxJS5qj6Yu1gpcOTVwi63GmiH47hJzVi4v2KYVGE%253d%26risl%3d%26pid%3dImgRaw%26r%3d0&exph=1200&expw=1920&q=fruit+basket&simid=608014842893978185&FORM=IRPRST&ck=70A49D7798309217C870739A36155F05&selectedIndex=18&itb=0>

Easttom, C. (2021). *Digital forensics, investigation, and response* (4th ed.). Jones & Bartlett Learning.

EmbeddedSW. (n.d.). *OpenPuff v4.01 Stenography & Watermarking* [pdf]. embeddedsw.net. Retrieved December 17, 2024, from <https://embeddedsw.net/doc/OpenPuff_Help_EN.pdf>

Kan, D., Jarmon, T., Kim, J., & Escobar, J. M. (2015, March 1). *Module 8: Steganography with S-Tools* [Video]. YouTube. <https://www.youtube.com/watch?v=eWDGBlmVSwA>

Luse, A. (2014, June 7). *OpenPuff Demo* [Video]. YouTube. <https://youtu.be/P1I8XXlNYoA?feature=shared>

Mixkit. (n.d.). *Incredible Free Sound Effects*. Retrieved December 17, 2024, from <https://mixkit.co/free-sound-effects/discover/trumpet-fanfare/>

ZanoList [ZanoList]. (204, October 5). *Hiding Secrets in Music with DeepSound (Steganography Explained)* [Video]. YouTube. <https://www.youtube.com/watch?v=pErS2qba5WY>

Appendix

Snapshots Of Evaluations Performed

Figure 1

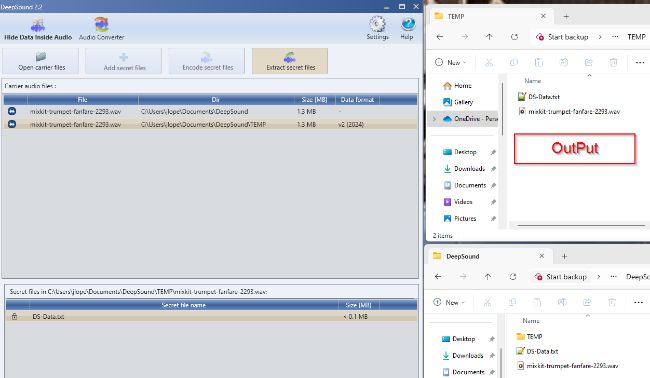


Figure 2

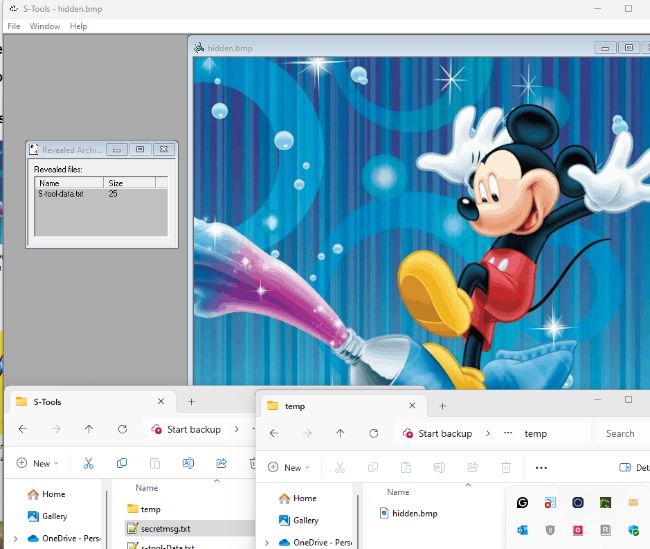


Figure 3

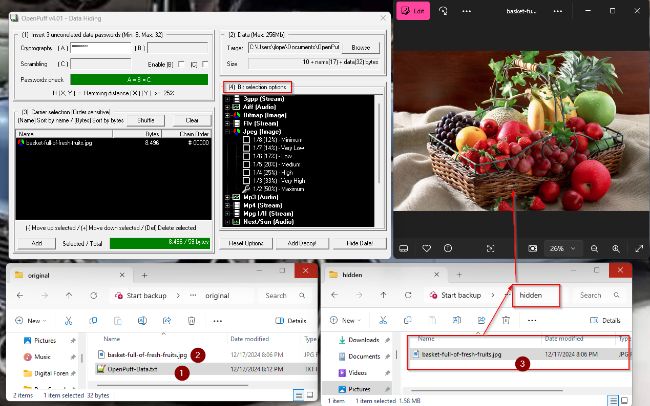


Figure 4

