SeungHwan Chung, Michael Gonzalez, Yevgeniya Martynyuk, Kenny Lin

CSCI400 Lab 1 Steganography

9/3/2018

Introductions

Team introduction and duties.

SeungHwan Chung: Leader, set up the environment, worked on embedding files, extracting, and gathered all the files and organize it. Also put all the information together for the problems.

Michael Gonzalez: Support leader, set up the environment, worked on Network tcp portion and analyzed it, chained embedded images, and calculation for capacity.

Yevgeniya Martynyuk: : Take a notes, analyzed the collected data, well organized it, and summarized into report.

Kenny Lin: Gather the notes and worked on Word Problems.

- Although we had an issue with communication, we end up worked together and contributed equally.

Steghide Process

We used a tool called “Steghide” to embed the given files into different cover files. We did Jpg to short text, long text, wav (sound file), and another jpg. Same thing to sound file to jpg, short text, long text, and another wav files. Same thing happens to short text and long text as a cover file with same files for embedding process. We eventually find out that Steghide tool only allows specific file formats such as WAV, JPEG, JPG, WAV, BMP, and AU. But it does not limits to embed file. So, we can use any embedding file types, but not with cover files.

In addition, cover file should have bigger size to fit embedding files. However, when we compare steganography file size to coverfile size, it only changes few kilobytes.

Embedding Data code

Steghide -cf(coverfile) CoverFile.(filetype) -ef(embedding file) EmbededFile.(filetype) -sf(savefile) NewFile.(filetype) -v

Cover File	Embed File	Steganography File	Passphrase	Successability.
jjc.jpg	secret.txt	new-imagetest_jjc.jpg	1234	True
jjc.jpg	longsecret.txt	new-longimagetest_jjc.jpg	1234	True
jjc.jpg	elephant_1.wav	new-songimagetest_jjc.jpg	1234	True
grand-steps.jpg	jjc.jpg	newgrand-steps.jpg	123	True
example.wav* (see the note1)	longsecret.txt	NewSummerLongSecret.wav	123	True
example.wav	secret.txt	NewSummerShortSecret.wav	123	True
example.wav	elephant_1.wav	NewSummerElephant.wav	123	True
example.wav	jordan_1.jpg	NewSummerJordan.wav	123	True
secret.txt	secret.txt	N/A	123	False*(see the note2)
secret.txt	longsecret.txt	N/A	123	False*(see the note2)
secret.txt	elephant_1.wav	N/A	123	False*(see the note2)
secret.txt	jjc.jpg	N/A	123	False*(see the note2)
longsecret.txt	secret.txt	N/A	123	False*(see the note2)
longsecret.txt	longsecret.txt	N/A	123	False
longsecret.txt	elephant_1.wav	N/A	123	False
longsecret.txt	jjc.jpg	N/A	123	False

Notes

1) Example.wav is SummerCricketsChirping.mp3. We used mpg123 tool to convert SummerCricketsChirping.mp3 to Example.wav.
    a) Code that we used:
        i) Sudo apt-get install mpg123 in order to download tools into kali linux.
        ii) Mpg123 -w example.wav(Converted files name creation) SummerCricketsChirping.mp3(mp3 file that is given).
2) Secret.txt file and longsecret.txt file type is txt file format. Steghide tool only supports WAV, JPEG, JPG, WAV, BMP, and AU as a coverfile. So, we can't use text file format as a cover file, but can use any file of format as
    a) http://steghide.sourceforge.net/documentation/manpage.php

Question for how would you embed information into a video (e.g. MPEG4)file? Give an example.

A. Since Steghide only provides specific file types as a cover file, we have no option but to follow the steghide manual. What we can do is video will be no longer a video. Which means, change the file format of MPEG4 to MP3 to WAV sound file format. Then we can use this WAV sound file format as a cover file to embed any file types of documents inside the sound file. If we use the video as a embedding file, then we can use any coverfile that has enough size to accept video.

Steghide Chaining Data

Embedding chain Data code

Steghide -cf(coverfile) CoverFile.(filetype) -ef(embedding file) EmbededFile.(filetype) -sf(savefile) NewFile.(filetype) -v

Steghide Chaining Data process is that we use one cover File and embedded the secret files multiple times which eventually end up one cover file with multiple hidden documents inside.

Cover File	Embedded File	Steganography File	Passcode	Sucess
grand-steps.jpg	longtext.txt	chainimg.jpg	123	True
chainimg.jpg	elephant_1.wav	chainimg2.jpg	123	True
chainimg.jpg	iron_man.jpg	chainimg2.jpg	123	True
chainimg3.jpg	jjc.jpg	chainimg4.jpg	123	True
chainimg4.jpg	bloodhound.jpeg	chainimg5.jpg	123	True

Steghide Information

Steghide info Newfile(Steganography file).(file type)

The capacity of the coverfile, was 225.1KB for each chain, and each chained embedding file had its own size attribution. For image to image particularly chainimg2.jpg (225.1kb) and iron_man.jpg (115.0kb) the percentage left to fill the cover file capacity was 1.96%. For image to audio using chainimg.jpg (225.1kb) and elephant_1.wav (7.7kb) the percentage left to fill the coverfile capacity was 29.23%. For image to text chainimg.jpg (225.1kb) and longtext.txt (2.7kb) the percentage left to fill the coverfile capacity was 83.37%.

Notes

1. There was an accidental additional chain added to the last step, the resulting chainimg was chainimg5.jpg.

Extracting Data file

This is an extracting data file from the steganography file to figure out what is embedded. We found that Chained files gave us error message with file format is not supporting. Therefore, regular steganography File with only one embedded file is extractable, but embedding multiple files will eventually corrupt the steganography file.

Extracting Data code

Steghide extract -sf new-longimagetest_jjc.jpgbb -xf longextractedtest.txt(creating extracting file) -v

Steganography File	Extracting File	Passphrase	Success
new-longimagetest_jjc.jpg	longextractedtest.txt	1234	True
chainimg5.jpg	extracted.txt	123	False*(see note1)
chainimg5.jpg	extracted.jpeg	123	True *(see note1)

Notes

1) Chainimg5.jpg can extract extracted.jpeg which is the last bloodhound.jpeg image file that we embedded into coverfile chainimg4.jpg. However, when we try to pull out either wav or txt files, it will still extract the data, but the file is already been corrupted with invalid characters. So, Chained extraction file could not be opened due to file format is not supported/corrupted.
    a) As we expected, extracted.jpeg showed us bloodhound.jpeg. We also expected extracted.txt will give us longtext.txt, but instead of longtext.txt, it gave us corrupted characters.

Embedding Data into Network Traffic

- Utilized the covert_tcp module for embedding network traffic.
- Covert_tcp has two mediums of establishing communication between two devices, device A being a receiver, device B being a sender.
- Each medium either sends a text file bit by bit within the packet headers or, interprets the headers, displays, and stores content of a text file.
- Once transmission was completed running ‘diff’ of the two files left us with a difference of 0, meaning the content of the files both sent and received were the same.
- If the data is interrupted during transmission using ctrl+c, the packets will be lost.
- However, if internet connection is lost, data transfer will stop too. This will be apparent in the pcap file.

Using tcpdump and or wireshark, we were able to monitor the packets which when analyzed through its .pcap file, enabled viewing of the encrypted the data transmission from our text file found within the headers of TCP/IP packets being sent through the network. This process is extremely useful for an attacker to send malicious traffic through a network in order to communicate (secretly) with another device inside the network.

Word Problems

1. Summarize the embedding techniques used by the tools.

The tools that we used, steghide, is to embed files that we want to hide into cover file. The cover file must have enough remaining space to embed the secret files. Steghide only offered specific file types as a cover file such as JPG, JPEG, and WAV.
Additionally, as mentioned earlier we also had to use mpg123 tool, a media converter in order to convert the crickets chirping audio to a .wav file as the .mp3 file format is not supported in steghide.

2. How would you detect the presence of steganography?

Steganalysis is the different methods to detect hidden information. These methods focus more on studying and analyzing on certain steganography methods. They try to detect and crack the information. It is hard to detect everyone because they each have their unique ways. These methods only work on steganography methods that they have already studied.
You can detect the presence of steganography by continuously looking for repetitive patterns. Once you find a repetitive patterns, you will be able to tell there are hidden informations in the image. Comparing the stego image with the original image will let you see the differences between them. Also you can use the Stegdetect tool to detect a steganographic image.

3. To what extent are the embedding techniques composable?

Embedding techniques are good for watermarks. People use steganography to put watermarks on their pictures to protects copyrights. People can also use the embedding technique communicate secretly to protect their personal information. Malicious users may send harmful

information using steganography. It would be difficult to detect these information without affecting the people who are using it just to protect their personal information.

4. How would you thwart steganographic efforts if you could be in the middle of the transmission,
i.e. You take the role of an active warden and modify traffic in transit?
If I could be in the middle of the transmission of two people trying to communicate each other. I would modify the message they send by replacing words with their synonyms without changing the meaning of the message. Doing this will most likely remove the hidden information that was trying to be sent.
You can also use the Discrete Spring Transform to remove potential dangerous hidden information and keep the rest of the data. This causes the numerical values of the image to be changed and the hidden information can't be recovered.

Conclusion

The purpose of this lab was to familiarize with steganography tools, perform a covert files analysis, conduct extraction and embeddelemet of various media files to demonstrate the importance of protection data on the network from corruption, attacks, or unauthorized access. Steganography uses cryptographic tools to encrypt files and insert them inconspicuously into different files; the goal is to make an encrypted data to appear as “noise” in case of a passive (listening) attack, so the secret information will travel on the network undetected. Conclusively, steganography is advantageous over cryptography because there is no way to tell that the encryption has occurred: the attacker will be unaware that the covert file was encrypted in the first place. This adds an extra layer of protection when supplemented with cryptographic encryption.

Throughout history steganography used fascinating methods to hide the facts of encryption; in current cyber security world its application is more functional as ever: it allows to hide water markings, passwords, keys, and any sensitive information. Today the most widely used technique is embedding hidden data into digital images (videos and pictures) using the least significant bit insertion, where the alterations is considerable undetected. In audio files embedding transpire when a weak inserted tone is masked by wide-band noise of the original message.

There are numerous methods to conduct steganalysis, where an extraction of embedded files occurs. To conduct this lab, in a controlled environment, passwords were used (when provided) as a method of detection of any alterations. Also covert.tcp was used to hide secret data into network traffic, running in Linux environment.

Ultimately, cryptography gives away the presence of data that is encrypted, while steganography does not arise any alerts of data being manipulated since it is hardly detectable.