Information Security (CP3404)

Chapter 5 practical

- This practical consists of some quick quiz questions, tutorial type questions, and hands-on projects relevant to (chosen from) Chapter 5 of the textbook (i.e., Mark Ciampa, *CompTIA® Security + Guide to Network Security Fundamentals, Fifth Edition*, USA, 2015).
- You may not be able to complete this practical within the scheduled one-hour practical session for this subject. You are strongly recommended to complete it in your own time (note that you are expected to work 10 hours per week on this subject, including 3 hours of contact time).

Section A – Quick-Quiz/Multiple-Choice Questions:

- Each week, complete the test on LearnJCU within the time allocated, which is before the next practical session.
- These on-line tests are worth 20% of the total marks for this subject.

Section B – Short Answer (Tutorial-Type) Questions:

- Answers to this set of questions are available at the end of this document (Section D).
- Effective learning implies you answer the questions before seeing the answer.

1. How can steganography be used to hide information in something other than images?
2. Discuss how cryptography can help ensure the availability of the data.
3. Explain hashing.
4. List and describe the characteristics a hashing algorithm must have to be considered secure.
5. Discuss how HMAC (Hashed Message Authentication Code) works.
6. Describe how Message Digest2 (MD2) works.
7. Describe how a block cipher works.
8. Describe hard disk drive encryption.

Section C – Hands-On Projects:

Due to security issues, you may not be allowed to practise hands-on projects with university's computers. Interested students are encouraged to do these projects on their own computers (if available). You will not be assessed for utilities/commands that cannot be practised on university computers. Note that you may still be assessed for descriptions/definitions that are provided in this section.

Using OpenPuff Steganography[1]

Unlike cryptography that scrambles a message so that it cannot be viewed, steganography hides the existence of data. In this project, you will use OpenPuff to create a hidden message.

1. Use your web browser to go to **embeddedsw.net/OpenPuff_Steganography_Home.html**[2].
2. Click **Source Page** and then click **Manual** to open the OpenPuff manual. save this file to your computer. Read through the manual to see the different features available.
3. Click your browser's back button to return to the home page.
4. Click **OpenPuff** to download the program.
5. Navigate to the location of the download and uncompress the Zip file on your computer.
6. Now Create a carrier file that will contain the hidden message. Open a Windows search box and enter **Snipping Tool**.
7. Launch **Snipping Tool**.
8. Under **New** click **Window Snip**.
9. Capture the image of one of the pages of the OpenPuff manual. Click **File** and **Save As**. Enter **Carrier1.png** and save to a location such as the desktop.
10. Now create the secret message to be hidden. Create a new Word file and enter **This is a secret message**.
11. Save this file as **Message.docx**.
12. Exit Word.
13. Create a Zip file from Message. navigate to the location of this file through Windows Explorer and click the right mouse button.
14. Click **Send to** and select **Compressed (zipped) folder** to create the Zip file.
15. Navigate to the OpenPuff directory and double-click **OpenPuff.exe**.
16. Click **Hide**.
17. Under (**1**) create three unrelated passwords and enter them into **Cryptography (A), (B),** and **(C)**.
18. Under (**2**) locate the message to be hidden. Click **Browser** and navigate to the file **Message.zip**. Click **Open**.
19. Under (**3**) select the carrier file. Click **Add** and navigate to **Carrier1.png** and click **Open** as shown in Figure 5-12.

[1] If you are concern about installing any of the software in this project on your regular computer, you can instead install the software in the Windows virtual machine created in practical-1. Software installed within the virtual machine will not impact the host computer.

[2] It is not unusual for websites to change the location of files. If the URL above no longer functions, open a search engine and search for "OpenPuff"

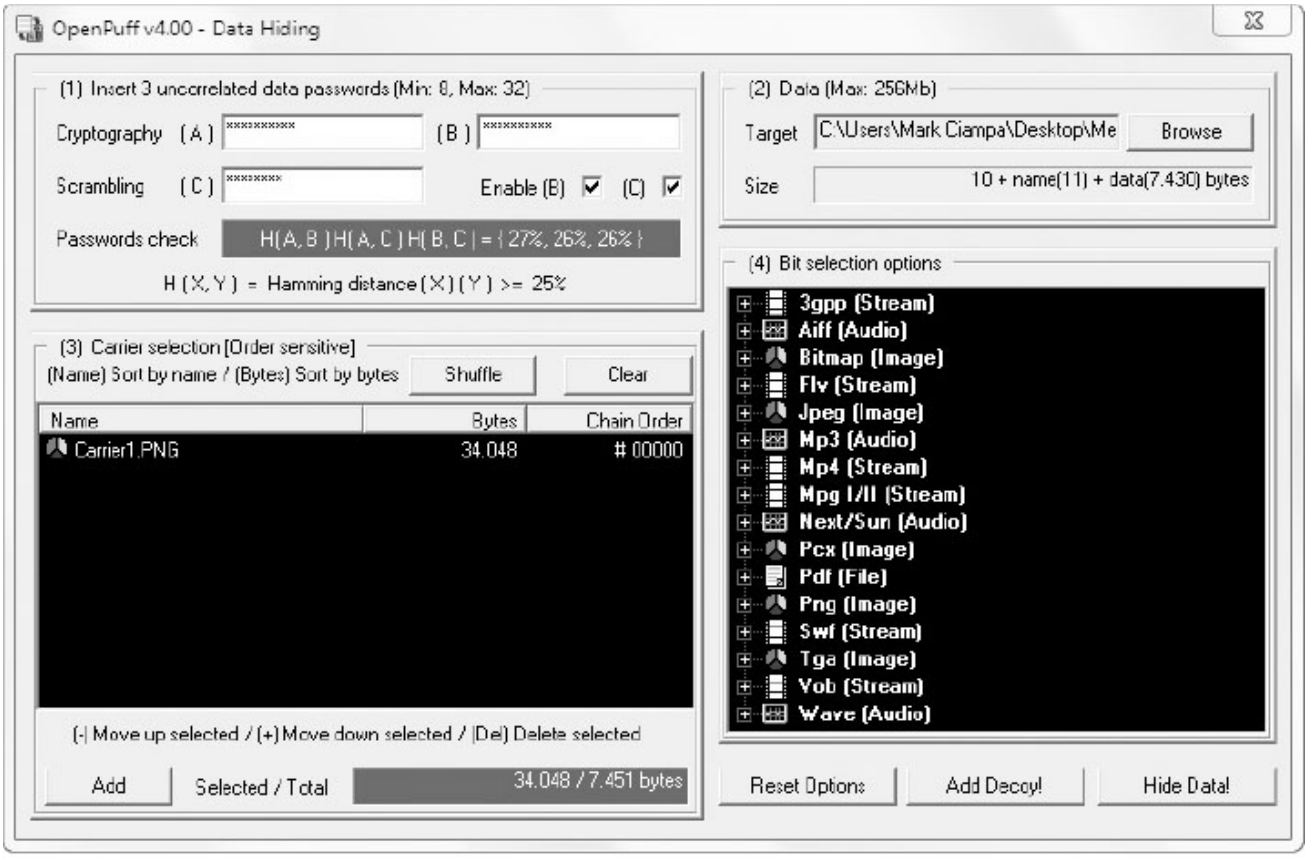

Figure 5-12 OpenPuff

Source: EmbeddedSW.net

20. Click **Hide Data!**.
21. Navigate to different location than that of the carrier files and click **OK**.
22. After the processing has completed, navigate to the location of the carrier file that contains the message and open the file. Can you detect anything different with the file now that it contains the message?
23. Now uncover the message. Close the OpenPuff data Hiding screen to return to the main menu.
24. Click **Unhide**.
25. Enter the three passwords.
26. Click **Add Carriers** and navigate to the location of **Carrier1** that contains the hidden message.
27. Click **Unhide!** and navigate to a location to deposit the hidden message. When it has finished processing click **OK**.
28. Click **done** after reading the report.
29. Go to that location and you will see **Message.zip**.
30. Close OpenPuff and close all windows.

Section D – Answers to Short Answer (Tutorial-Type) Questions:

1. How can steganography be used to hide information in something other than images?

   Answer:
   Steganography can be used to hide data within the file header fields that describe the file, between sections of the metadata.

2. Discuss how cryptography can help ensure the availability of the data.

   Answer:
   Rather than keeping an important file or data stored on a secured hard drive, an encrypted file can be made available from a centralized location, and only individuals in possession of the key can decrypt the information.

3. Explain hashing.

   Answer:
   Hashing is a process for creating a unique digital fingerprint for a set of data. This fingerprint, called a hash (sometimes called a one-way hash or digest) represents the contents. Although hashing is considered a cryptographic algorithm, its purpose is not to create a ciphertext that can later be decrypted. Instead, hashing is one-way in that its contents cannot be used to reveal the original set of data. Hashing is primarily used for comparison purposes.

4. List and describe the characteristics a hashing algorithm must have to be considered secure.

   Answer: *Fixed size* – A hash of a short set of data should produce the same size as a hash of a long set of data. For example, a hash of the single letter a is 86be7afa339d0fc7cfc785e72f578d33, while a hash of 1 million occurrences of the letter a is 4a7f5723f954eba1216c9d8f6320431f, the same length.

   Unique – Two different sets of data cannot produce the same hash, which is known as a collision. Changing a single letter in one data set should produce an entirely different hash. For example, a hash of Today is Tuesday is 8b9872b8ea83df7152ec0737d46bb951 while a hash of today is Tuesday (changing the initial T to t) is 4ad5951de752ff7f579a87b86bfafc2c.

   Original – It should be impossible to produce a data set that has a desired or predefined hash.

   Secure – The resulting hash cannot be reversed in order to determine the original plaintext.

5. Discuss how HMAC (Hashed Message Authentication Code) works.

   Answer:
   HMAC begins with a shared secret key that is in the possession of both the sender and receiver. The sender creates a hash and then encrypts that hash with the key before transmitting it with the original data. The receiver uses their key to decrypt the hash and then creates their own hash of the data, comparing the two values.

6. Describe how Message Digest2 (MD2) works.

   Answer:
   Message Digest 2 (MD2) takes plaintext of any length and creates a hash 128 bits long. MD2 begins by dividing the message into 128-bit sections. If the message is fewer than 128 bits, data known as padding is added. For example, if a 10-byte message is abcdefghij, MD2 would pad the message to make it abcdefghij666666 to create a length of 16 bytes (128 bits). The padding is always the number of bytes that must be added to create a length of 16 bytes; in this example, 6 is the padding because 6 more bytes had to be added to the 10 original bytes. After padding, a 16-byte checksum is appended to the message.

7. Describe how a block cipher works.

   Answer:
   A block cipher manipulates an entire block of plaintext at one time. The plaintext message is divided into separate blocks of 8 to 16 bytes, and then each block is encrypted independently. For additional security, the blocks can be randomized.

8. Describe hard disk drive encryption.

Answer:
Just as an encrypted hardware-based USB flash drive will automatically encrypt any data stored on it, self-encrypting hard disk drives (HDDs) can also protect all files stored on them. When the computer or other device with a self-encrypting HDD is initially powered up, the drive and the host device perform an authentication process. If the authentication process fails, the drive can be configured to simply deny any access to the drive or even perform a cryptographic erase on specified blocks of data (a cryptographic erase deletes the decryption keys so that all data is permanently encrypted and unreadable). This also makes it impossible to install the drive on another computer to read its contents.