cs3235:3: - Laboratory #3 for September 11th, 2019

The writeup for this laboratory[1] can be done in groups of two, but the first two parts should be done individually. The learning objective of this lab is for students to get familiar with the concepts in the secret-key encryption. After finishing the lab, students should be able to gain a first-hand experience on encryption algorithms, encryption modes, paddings, and initial vector (IV). Moreover, students will be able to use tools and write programs to encrypt/decrypt messages.

1 Lab 3, part 1: WSE

Today we have a case study of the Windows Scripting Encoder. Even simple encryption schemes still find use today! In particular, we will investigate a slightly simplified version of the Windows Scripting Encoder, provided by Microsoft a few years ago to "encrypt" the program code of programs running on webservers. The scripting engine itself could decode and execute the program code, but it looks encrypted to anyone else. The motivation behind the Scripting Encoder was to prevent an attacker who illegally downloaded these programs from gaining any information about how the program works. Quite often programs written from the web also contain passwords for database servers hidden among the program code. The Scripting Encoder's aim was to hide these from an attacker.

In this part of the laboratory, your goal is to investigate how it works and to recover a password!

Firstly, open a terminal window in Unix on the laboratory machine. In your home directory on the laboratory machine, create some test data files to be encrypted. Each file can have a small amount of text in it (sample passwords in, say, `sample1.txt`, `sample2.txt`...). Use `wse` to encrypt them:

```
wse < sample1.txt
```

Alternatively, you can just enter in strings directly from the console window "`wse`". The password for the cs3235 grading administrator has been found by harriet-the-hacker in a wse-encrypted file. You can get it by using wget, a useful tool for retreiving files:

```
wget https://hugh.comp.nus.edu.sg/cs3235/lab3/adminpassword
```

[1] Part 1 adapted from Stephane Werner's (NUS) lab, and the rest is a SEED lab...

Look at the password file by typing "`cat adminpassword`". You will recognize it as a wse-encrypted version of the password. If you can discover the original administrator password, you can input your grade through the web page, as before. Your goal is to figure out how the encoding works so you can learn the password!

During the lecture, we discussed a few possible attacks against encryption systems. Think about the chosen ciphertext and chosen plaintext attacks. Perhaps one of them would be helpful to you here. Now it's time to investigate! You can try to encrypt as many messages as you want by running the "`wse`" program.

Access the cs3235 grading website, and enter your username, password, and the adminpassword, along with whatever you want to give yourself as a mark:

`https://hugh.comp.nus.edu.sg/cs3235/lab3/gradeslab3-1.php`

You can give yourself whatever mark you believe you will in the future deserve.

2 Lab 3, part 2: Encryption and decryption using different ciphers

The learning objective of this lab is for students to get familiar with the concepts in the secret-key encryption. After finishing the lab, students should be able to gain a first-hand experience on encryption algorithms, encryption modes, paddings, and initial vector (IV). Moreover, students will be able to use tools and write programs to encrypt/decrypt messages.

In this task, we will play with various encryption algorithms and modes. You can use the following `openssl enc` command to encrypt/decrypt a file. To see the manuals, you can type `man openssl` and `man enc`. In the following command, replace the `ciphertype` with a specific cipher type, such as `-aes-128-cbc`, `-aes-128-cfb`, `-bf-cbc`, etc.

```
openssl enc ciphertype -e  -in plain.txt -out cipher.bin\
    -K 00112233445566778889aabbccddeeff -iv 0102030405060708
```

What size of key is being used above? In this task, you should try at least 3 different ciphers and three different modes. You can find the meaning of the command-line options and all the supported cipher types by typing "`man enc`". We include some common options for the `openssl enc` command in the following:

```
-in <file>    input file
-out <file>   output file
-e            encrypt
-d            decrypt
-K/-iv        key/iv in hex is the next argument
```

Once you are familiar with this use of the `openssl` program, retrieve an encrypted file from the server:

```
wget https://hugh.comp.nus.edu.sg/cs3235/lab3/lab3-2.bin
```

This is an encrypted message, using the exact same key and IV given above. Decrypt it - it is a secret English phrase you can use to give yourself your heart's desire!

Access the cs3235 grading website (you know the drill by now):
`https://hugh.comp.nus.edu.sg/cs3235/lab3/gradeslab3-2.php`
You can give yourself whatever mark you believe you will in the future deserve.

2.1 Writeup task 1: Encryption Mode – ECB vs. CBC

The file `https://hugh.comp.nus.edu.sg/cs3235/lab3/pic_original.bmp` contains a simple picture. We would like to encrypt this picture, so people without the encryption keys cannot know what is in the picture. Please encrypt the file using the ECB (Electronic Code Book) and CBC (Cipher Block Chaining) modes, and then do the following:

1. Let us treat the encrypted picture as a picture, and use a picture viewing software to display it. However, For the `.bmp` file, the first 54 bytes contain the header information about the picture, we have to set it correctly, so the encrypted file can be treated as a legitimate `.bmp` file. We will replace the header of the encrypted picture with that of the original picture. You can use a hex editor tool (e.g. `ghex` or `Bless`) to directly modify binary files.
2. Display the encrypted picture using any picture viewing software. Can you derive any useful information about the original picture from the encrypted picture?

2.2 Writeup task 2: Encryption Mode – Corrupted Cipher Text

To understand the properties of various encryption modes, we would like to do the following exercise:

1. Create a text file that is at least 64 bytes long.
2. Encrypt the file using the AES-128 cipher.
3. Unfortunately, a single bit of the 30th byte in the encrypted file got corrupted. You can achieve this corruption using a hex editor.
4. Decrypt the corrupted file (encrypted) using the correct key and IV.

Please answer the following questions:

1. How much information can you recover by decrypting a corrupted file, if the encryption mode is ECB, CBC, CFB, or OFB, respectively?
2. Please explain why.
3. What are the implication of these differences?

3 Lab 3, part 3: Submission of your (extra) writeup tasks

Your task: write a short (less than 8000 characters) lab report to describe what you have done and what you have observed, in particular - for the "Writeup Tasks" 1 and 2. For task 1, you should show your results, and explain what you are seeing. For task 2, you should answer the three questions in as concise a way as you can.

Once you have your plan ready, both you and your lab partner should access the cs3235 grading website to enter in your plan:

`https://hugh.comp.nus.edu.sg/cs3235/lab3/gradeslab3-3.php`

You will only see your own plan, not everyone else's. The laboratory will remain online and open until Saturday midnight.