M Tech I : Introduction to Computer Security

Autumn Quarter 2017-18

Lab Assignment 1

Date of Submission : 4th OCT 2017

1. Write a a menu driven program for Shift cipher with following functions : (a) Encrypt given plain text. (b). Decrypt given cipher text. (c). Find encryption key using brute force attack. (d). Find encryption key using frequency analysis attack. Consider file as an input in program. The program should work for large and variable length input text.

   The following assignments now concern the Stack Analysis and demonstrating various Vulnerabilities in C.

2. Assume that a function my_func() is called by the main function main(). Assume the initial structure of the stack as shown below - just before the call to the function my_func().

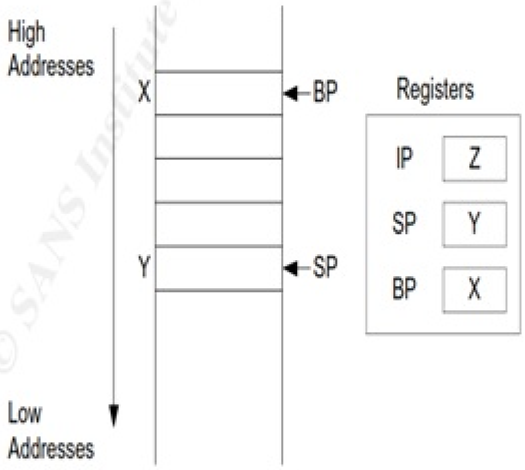

   Fig. The Snapshot of Stack before call to my_func()

3. Write a program viz. prompt-name that uses a function viz. “display-name()”. The main function should be invoked as *$ prompt-name MYNAME* i.e. with the first name (only one word) of the person as argument. The program then should pass on this argument to the function display-name, which should print the name passed as argument on the screen and return to main. The main should print “Name Displayed and over”and end. Analyze the state of stack and related registers at the following breakpoints:

   (a) At the time of calling the function in main.

   (b) After the prolog of function called.

   (c) At the return statement of the called function.

   (d) At the execution of the last statement in main.

4. Why is subtraction performed in the instruction at the address $0x0804840e$? Identify the instructions for the prolog and for the epilog in this code dump.

Listing 1: Problem 2

```
(gdb) disas add
Dump of assembler code for function add:
0x0804840b <+0>:        push    %ebp
0x0804840c <+1>:        mov     %esp,%ebp
0x0804840e <+3>:        sub     $0x10,%esp
0x08048411 <+6>:        movl    $0x0,-0x4(%ebp)
0x08048418 <+13>:       mov     0x8(%ebp),%edx
0x0804841b <+16>:       mov     0xc(%ebp),%eax
0x0804841e <+19>:       add     %edx,%eax
0x08048420 <+21>:       mov     %eax,-0x4(%ebp)
0x08048423 <+24>:       mov     -0x4(%ebp),%eax
0x08048426 <+27>:       leave
0x08048427 <+28>:       ret
End of assembler dump.
```

5. Write a menu driven program with appropriate functions to implement the affine cipher i.e. $c = am + b(mod 26)$. Let the values of a and b be entered by the user. Your program must check for the feasibility of these values before encrypting the plaintext m. The program must also output the decrypted values of the plain text. LEt the plaintext m be input as an character array of defined size.

6. Write a program to implement the Playfair cipher using matrix specifications as entered by the user (maximum 6x6). Let the key and the plaintext also be interactively entered by the user.

7. Write a program to implement the columnar transposition cipher.

8. Write a program to implement rail fence transposition cipher. Write a program for Encryption using key (depth). Write a program for decryption using key (depth). Your program must also illustrate an approach to cryptanalyze the cipher and print the key obtained after cryptanalysis.

9. Write a program to implement Vernam cipher - Generate key(s), read plain text/cipher text from a file, read key (from file), write output to a file - with the following specifications :

   (a) Generate Key$s$ - generate stream of $1's$ and $0's$ using Random Number Generator, group them and write as binary code in a file.

   (b) Read plain text - plain text (ASCII) from file can be first converted in binary code. If your key is shorter than plain text - repeat the key.

   (c) Encryption: Plaintext XOR Key.

   (d) Write output - write cipher text in binary code in a file

   (e) Decryption - Ciphertext XOR Key

   (f) Write Output - Convert decrypted binary code in ASCII and write in a file

10. Write a program to implement n-gram Hill Cipher. Read encryption key/decryption key, Read plain text/cipher text, give output.

11. Write a program to implement the Vigenre Cipher. Your program must work interactively asking the user to input the key and the plaintext/ciphertext and the mode of operation (encrypt/decrypt). The plaintext/ciphertext entered must be a sentence with maximum 120 alphanumeric characters. The program then must encrypt/decrypt the plaintext/ciphetext and the display the output.

12. Write a program to illustrate the Kasiski's method for cryptanalysis of the ciphertext obtained using the Vigenre cipher. The program must get the ciphertext from the user interactively and use appropriate algorithmic approach to identify the repeated characters in the ciphertext.

13. Write a program that takes ciphertext as input and computes the Index of Coincidence (as in Friedman's text).