HW #2: Security Audit

CS5950 Fall 2015

Due: *Monday, October 12 @ 11:59pm*

Perform a security audit on the following program. On a separate sheet, list all security related vulnerabilities that you find.

```
/*
* This program encrypts data in source file and puts it in
* the destination file.
* Use: Encrypt sourceFile destinationFile key
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#define ALPHABET_SIZE 26
#define ALPHABET "abcdefghijklmnopqrstuvwxyz"
#define NEW_LINE_ASCII 10
#define ENCRYPT 1
#define DECRYPT 0
int encryptCaesar(int character, int shift, int encryptOrDecrypt)
{
    if(character ==NEW_LINE_ASCII)
        return character;
    const char alphabet[ALPHABET_SIZE*2] = ALPHABET ALPHABET;
    return (encryptOrDecrypt==ENCRYPT
        ? alphabet[character-'a'+shift]
         : alphabet[character-'a'-shift]);
}

char* capitalizeWord(char* word)
{
    int i =0;
    char* capitalized = malloc(sizeof(char)*strlen(word));
    if(capitalized ==NULL) exit(1);
    while(i<strlen(word))
    {
        capitalized[i] = toupper(word[i]);
        i++;
    }
    return strdup(capitalized);
}
char* encryptVigenere(char* string, char* password, int encodeOrDecode)
{
```

```
    int i, keyLength, stringLength;
    char *dest = malloc(sizeof(char)*strlen(string));
    if(dest==NULL) exit(1);
    dest = strdup(string);
    dest = capitalizeWord(dest);
    password = capitalizeWord(password);

    /* strip out non-letters */
    for (i = 0, stringLength = 0; dest[stringLength] != '\0'; stringLength++)
            if (isupper(dest[stringLength]))
                    dest[i++] = dest[stringLength];

    keyLength = strlen(password);
    for (i = 0; i < stringLength; i++) {
          if (!isupper(dest[i])) continue;
                dest[i] = 'A' + (encodeOrDecode
                    ? dest[i] - 'A' + password[i % keyLength] - 'A'
                    : dest[i] - password[i % keyLength] + 26) % 26;
    }

    return dest;
}

void main (int argc, char *argv[])
{
    char message[300];
    sprintf(message, "Program Name: %s \n", argv[0]);
    printf(message);
    char c;
    char*  sourceFile = malloc(sizeof(char)*50);
    if(sourceFile==NULL) exit(1);
    char* destinationFile = malloc(sizeof(char)*50);
    if(destinationFile==NULL) exit(1);
    int shift;
    FILE * dFile;
    FILE * sFile;

    char inputConfirmation[200];
    sprintf(inputConfirmation, "You entered %.50s as the input and %.50s as the output \n",
            argv[1], argv[2]);
    sprintf(message, inputConfirmation);
    printf(message);
    strcpy(sourceFile, argv[1]);
    strcpy(destinationFile, argv[2]);
    if(access(sourceFile, W_OK)!=0) exit(1);
```

```
    if(access(destinationFile, W_OK)!=0) exit(1);


    printf("1.     Encrypt using Caesar's Cipher\n");
    printf("2.     Decrypt using Caesar's Cipher\n");
    printf("3.     Encrypt using Vigenere Cipher\n");
    printf("4.     Decrypt using Vigenere Cipher\n");
    printf("Select Task Number: ");
    int choice;
    scanf("%d", &choice);

    sFile = fopen(sourceFile, "r");
    dFile = fopen(destinationFile, "w");
    if(choice ==1 || choice==2)
    {
        int encryptOrDecrypt = choice==1?ENCRYPT:DECRYPT;
        char* shiftPrompt = "Enter shift: ";
        printf(shiftPrompt);
        scanf("%d", &shift);
        shift = shift- (ALPHABET_SIZE*(shift / ALPHABET_SIZE));
        while((c=getc(sFile))!=EOF)
        {
            char t = encryptCaesar(tolower(c),shift, encryptOrDecrypt);
            if(isupper(c))
                t=toupper(t);
            putc(t, dFile);
        }
    }

    else if (choice==3|| choice==4)
    {
        int encryptOrDecrypt = choice==3?ENCRYPT:DECRYPT;
        printf("Enter an 8 letter password:\n");
        char password[8];
        gets(password);

        char buf[256];
          while (fgets (buf, sizeof(buf), sFile)) {
               fprintf(dFile, encryptVigenere(buf, &password[0], encryptOrDecrypt));
        }
    }
    return;

}
```