

Forum:[Hack Forums](#)

Thread: [What's wrong with my Encryption thingy??](#)

Consulting Detective - March 3, 2020, 3:08 am

Hi guys. I'm messing around with file encryption in C. I wrote a small program that will take a file, process it through 3 encryption algorithms and creates a new file with the encrypted output.

My problem is that after the string in the buffer goes through the first algorithm, it is not giving the right output I want.

Output:

PHP Code

```
<?php
```

```
Thequickbrownfoxjumpsoverthelazydog. //buffer
```

```
a?L //result of first algorithm
```

```
u?L
```

```
0?g
```

PHP Code

```
<?php
```

```
#include<stdio.h>
```

```
#include<string.h>
```

```
#define XOR_KEY1 0x2B //nice
```

```
#define XOR_KEY2 0x45 //golden rule
```

```
int main(int argc, char *argv[]) {
```

```
    char buff[1024];
```

```
    char placeHolder;
```

```
    int len = strlen(buff);
```

```
    int x;
```

```
    FILE *sourceFile;
```

```
    FILE *destFile;
```

```
    sourceFile = fopen(argv[1], "r");
```

```
    destFile = fopen("encryptedFile.txt", "a");
```

```
    while(feof(sourceFile) == 0) {
```

```
        fgets(buff, sizeof(buff), sourceFile);
```

```
        puts(buff);
```

```
//prints initial content of buffer
```

```
        for(x = 0; buff[x] != '\0'; x++, len--) {
```

```
//1st obfuscation
```

```
            placeHolder = buff[len-1];
```

```
            buff[len-1] = buff[x];
```

```
            buff[x] = placeHolder;
```

```

    if(&buff[x] == &buff[len-1] || &buff[x+1] == &buff[len-1]) {
        break;
    }
}

puts(buff); //prints output of 1st

for(x = 0; buff[x] != '\0'; x++) { //2nd obfuscation
    switch(buff[x]) {
        case 'a':
            buff[x] = 'u';
            break;
        case 'e':
            buff[x] = 'o';
            break;
        case 'o':
            buff[x] = 'e';
            break;
        case 'u':
            buff[x] = 'a';
            break;
        case 'A':
            buff[x] = 'U';
            break;
        case 'E':
            buff[x] = 'O';
            break;
        case 'O':
            buff[x] = 'E';
            break;
        case 'U':
            buff[x] = 'A';
            break;
        default:
            break;
    }
}

puts(buff); //prints output of 2nd

for(x = 0; buff[x] != '\0'; x++) { //3rd obfuscation
    if(buff[x] == 'O') {
        continue;
    }
    if((x % 2 == 0) && (x != 0)) {
        buff[x] = buff[x] ^ XOR_KEY1;
        continue;
    }
    buff[x] = buff[x] ^ XOR_KEY2;
}

puts(buff); //prints output of 3rd

```

```

        fprintf(destFile, "%s", buff);
    }

    fclose(sourceFile);
    fclose(destFile);

    return 0;

}

```

P.S. sorry for the messy code, still learning.

Zecture - March 3, 2020, 3:44 am

(Mar 3, 2020 04:08 AM)Consulting Detective Wrote: [\[CITE\]](#) Hi guys. I'm messing around with file encryption in C. I wrote a small program that will take a file, process it through 3 encryption algorithms and creates a new file with the encrypted output.

My problem is that after the string in the buffer goes through the first algorithm, it is not giving the right output I want.

Output:

PHP Code

```
<?php
```

```
Thequickbrownfoxjumpsoverthelazydog. //buffer
```

```
a?L //result of first algorithm
```

```
u?L
```

```
0?g
```

PHP Code

```
<?php
```

```
#include<stdio.h>
```

```
#include<string.h>
```

```
#define XOR_KEY1 0x2B //nice
```

```
#define XOR_KEY2 0x45 //golden rule
```

```
int main(int argc, char *argv[]) {
```

```
    char buff[1024];
```

```
    char placeHolder;
```

```
    int len = strlen(buff);
```

```
    int x;
```

```
    FILE *sourceFile;
```

```
    FILE *destFile;
```

```
    sourceFile = fopen(argv[1], "r");
```

```
    destFile = fopen("encryptedFile.txt", "a");
```

```
    while(feof(sourceFile) == 0) {
```

```
        fgets(buff, sizeof(buff), sourceFile);
```

```
puts(buff); //prints initial content of buffer
```

```
for(x = 0; buff[x] != '\0'; x++, len--) { //1st obfuscation  
    placeHolder = buff[len-1];  
    buff[len-1] = buff[x];  
    buff[x] = placeHolder;  
    if(&buff[x] == &buff[len-1] || &buff[x+1] == &buff[len-1]) {  
        break;  
    }  
}
```

```
puts(buff); //prints output of 1st
```

```
for(x = 0; buff[x] != '\0'; x++) { //2nd obfuscation  
    switch(buff[x]) {  
        case 'a':  
            buff[x] = 'u';  
            break;  
        case 'e':  
            buff[x] = 'o';  
            break;  
        case 'o':  
            buff[x] = 'e';  
            break;  
        case 'u':  
            buff[x] = 'a';  
            break;  
        case 'A':  
            buff[x] = 'U';  
            break;  
        case 'E':  
            buff[x] = 'O';  
            break;  
        case 'O':  
            buff[x] = 'E';  
            break;  
        case 'U':  
            buff[x] = 'A';  
            break;  
        default:  
            break;  
    }  
}
```

```
puts(buff); //prints output of 2nd
```

```
for(x = 0; buff[x] != '\0'; x++) { //3rd obfuscation  
    if(buff[x] == 'O') {  
        continue;  
    }  
    if((x % 2 == 0) && (x != 0)) {  
        buff[x] = buff[x] ^ XOR_KEY1;  
        continue;  
    }  
}
```

```
}  
buff[x] = buff[x] ^ XOR_KEY2;  
}  
  
puts(buff); //prints output of 3rd  
  
fprintf(destFile, "%s", buff);  
}  
  
fclose(sourceFile);  
fclose(destFile);  
  
return 0;  
  
}
```

P.S. sorry for the messy code, still learning.

You're getting the length of the string before reading it from the file and writing it into the buffer

Consulting Detective - March 3, 2020, 3:50 am

(Mar 3, 2020 04:44 AM)Zexture Wrote: [\[CITE\]](#) You're getting the length of the string before reading it from the file and writing it into the buffer

Oh you're right duh. Thank you :)