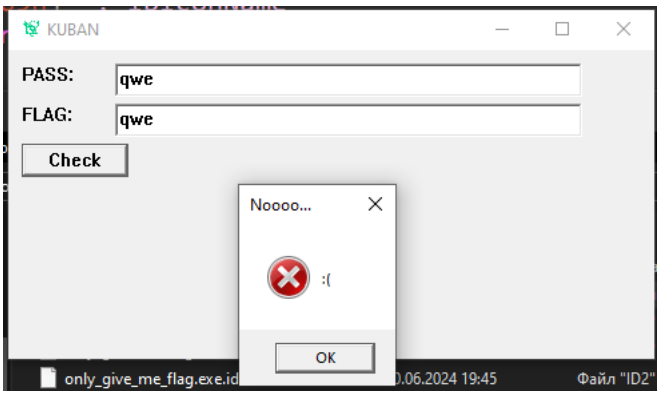
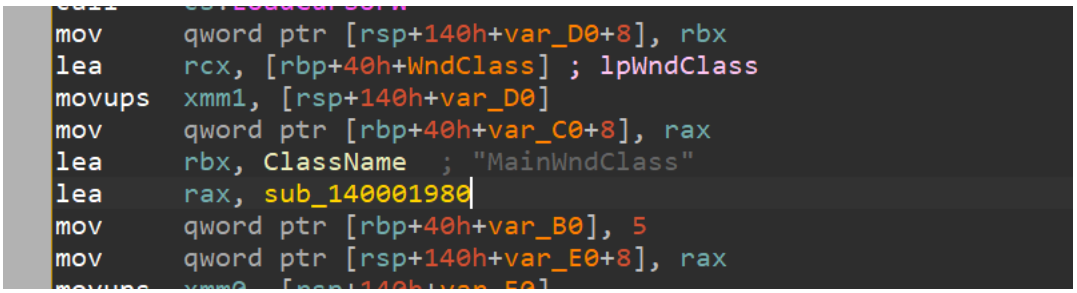
Название:	Только дай мне флаг
Категория:	Реверс
Уровень:	Средний
Очки:	400
Описание:	Хей, ревёрсер, ты просто введи флаг и всё. А кодовую фразу ты знаешь.
Теги:	AES, Побайтовое шифрование, C++, WinForms, KubanCTF
Автор:	ROP

Прохождение:

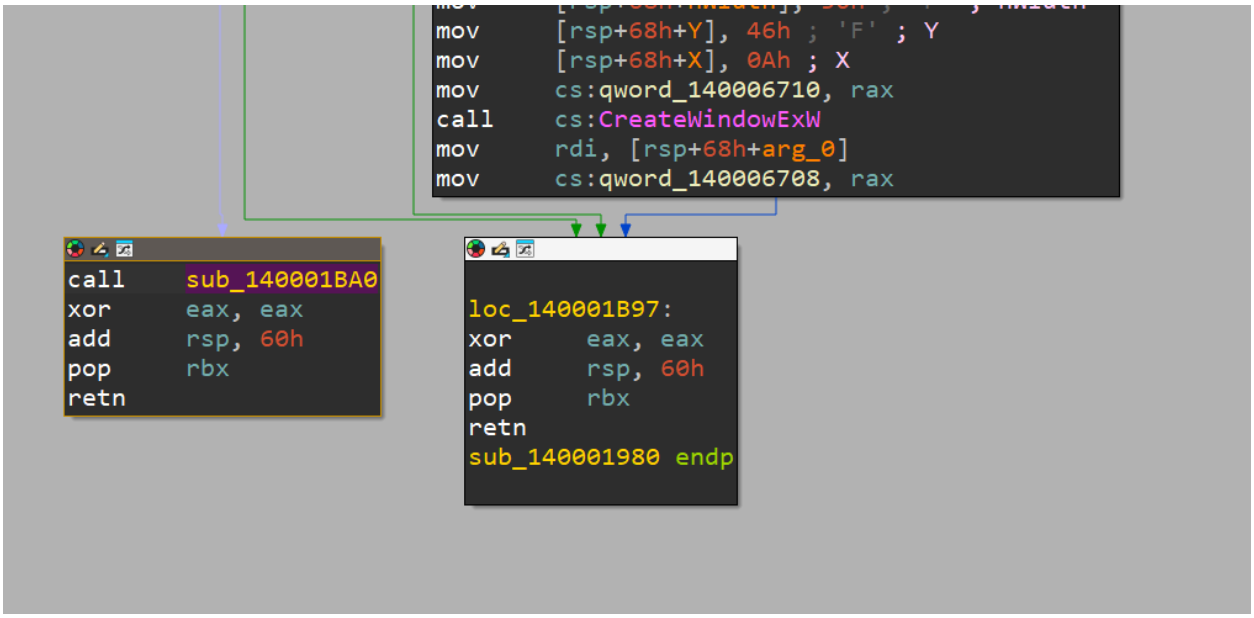
Изучаем исполняемый файл.



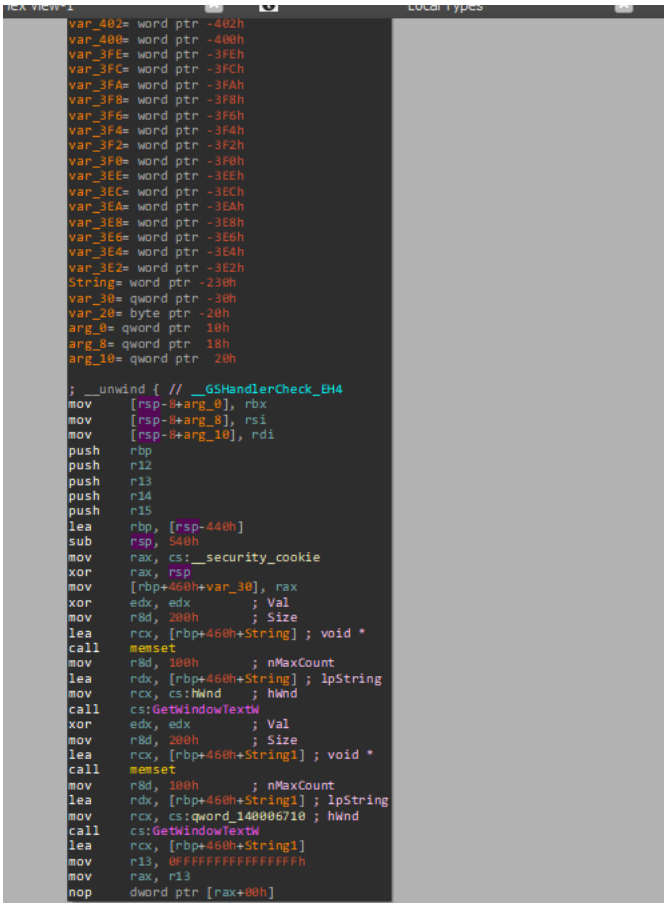
Смотрим в IDA. Это плюсы и формы.



Переходим в эту функцию.


mov [rsp+68h+Y], 46h ; 'F' ; Y
mov [rsp+68h+X], 0Ah ; X
mov cs:qword_140006710, rax
call cs:CreateWindowExW
mov rdi, [rsp+68h+arg_0]
mov cs:qword_140006708, rax
call sub_140001BA0
xor eax, eax
add rsp, 60h
pop rbx
retn
loc_140001B97:
xor eax, eax
add rsp, 60h
pop rbx
retn
sub_140001980 endp

Затем сюда.



```
var_402= word ptr -402h
var_400= word ptr -400h
var_3FE= word ptr -3FEh
var_3FC= word ptr -3FCh
var_3FA= word ptr -3FAh
var_3F8= word ptr -3F8h
var_3F6= word ptr -3F6h
var_3F4= word ptr -3F4h
var_3F2= word ptr -3F2h
var_3F0= word ptr -3F0h
var_3EE= word ptr -3EEh
var_3EC= word ptr -3ECh
var_3EA= word ptr -3EAh
var_3E8= word ptr -3E8h
var_3E6= word ptr -3E6h
var_3E4= word ptr -3E4h
var_3E2= word ptr -3E2h
String= word ptr -230h
var_30= qword ptr -30h
var_20= byte ptr -20h
arg_0= qword ptr  10h
arg_8= qword ptr  18h
arg_10= qword ptr  20h

; __unwind { // __GSHandlerCheck_EH4
mov     [rsp-8+arg_0], rbx
mov     [rsp-8+arg_8], rsi
mov     [rsp-8+arg_10], rdi
push    rbp
push    r12
push    r13
push    r14
push    r15
lea     rbp, [rsp-440h]
sub     rsp, 540h
mov     rax, cs:__security_cookie
xor     rax, rsp
mov     [rbp+460h+var_30], rax
xor     edx, edx        ; Val
mov     r8d, 200h       ; Size
lea     rcx, [rbp+460h+String] ; void *
call    memset
mov     r8d, 100h       ; nMaxCount
lea     rdx, [rbp+460h+String] ; lpString
mov     rcx, cs:hWnd    ; hWnd
call    cs:GetWindowTextW
xor     edx, edx        ; Val
mov     r8d, 200h       ; Size
lea     rcx, [rbp+460h+String1] ; void *
call    memset
mov     r8d, 100h       ; nMaxCount
lea     rdx, [rbp+460h+String1] ; lpString
mov     rcx, cs:qword_140006710 ; hWnd
call    cs:GetWindowTextW
lea     rcx, [rbp+460h+String1]
mov     r13, 0FFFFFFFFFFFFFFFFh
mov     rax, r13
nop     dword ptr [rax+00h]
```

Тут уже основной код проверки:

```
#define LEN_PASS 40


    wchar_t login[256] = { 0 };
    GetWindowText(hLoginEdit, login, sizeof(login) / sizeof(login[0]));

    wchar_t password[256] = { 0 };
    GetWindowText(hPasswordEdit, password, sizeof(password) / sizeof(password[0]));


    bool passwordMatch = true;
    if (wcslen(password) != LEN_PASS || wcsncmp(password, L"CSC{", 4) != 0 || password[LEN_PASS
        passwordMatch = false;
    }
    else {
        int dashCount = 0;
        for (int i = 0; i < LEN_PASS; ++i) {
            if (password[i] == L'-') {
                dashCount++;
            }
```

```
        }
        if (dashCount != 3)
            passwordMatch = false;
        for (int i = 12; i < LEN_PASS-1; i += 9) {
            if (password[i] != L'-' || !iswxdigit(password[i + 1]) || !iswxdigit(password[i + 2]
                passwordMatch = false;
                break;
            }
        }
    }

    unsigned long part1;
    unsigned long part2;
    unsigned long part3;
    unsigned long part4;


    if (passwordMatch) {
        wchar_t* endPtr;


        part1 = wcstoul(&password[4], &endPtr, 16);
        if (*endPtr != L'-') {
            passwordMatch = false;
        }

        part2 = wcstoul(&password[13], &endPtr, 16);
        if (*endPtr != L'-') {
            passwordMatch = false;
        }

        part3 = wcstoul(&password[22], &endPtr, 16);
        if (*endPtr != L'-') {
            passwordMatch = false;
        }

        part4 = wcstoul(&password[31], &endPtr, 16);
        if (*endPtr != L'}') {
            passwordMatch = false;
        }
    }

    struct AES_ctx ctx;

    const unsigned char key[16] = {
    0xD6, 0x23, 0xB8, 0xEF, 0x62, 0x26, 0xCE, 0xC3, 0xE2, 0x4C, 0x55, 0x12,
    0x7D, 0xE8, 0x73, 0xE7
    };

    const unsigned char iv[16] = {
    0x18, 0x42, 0x31, 0x2D, 0xFC, 0xEF, 0xDA, 0xB6, 0xB9, 0x49, 0xF1, 0x0D,
    0x03, 0x7E, 0x7E, 0xBD
    };
    AES_init_ctx_iv(&ctx, key, iv);
```

```
    size_t len = wcslen(login);
    std::vector<uint8_t> login_buffer(len * sizeof(wchar_t) + 1); // +1 для нулевого символа
    size_t convertedChars = 0;
    wcstombs_s(&convertedChars, reinterpret_cast<char*>(login_buffer.data()), login_buffer.size(


    login_buffer.resize(convertedChars - 1);

    addPKCS7Padding(login_buffer, 16);




    AES_CBC_encrypt_buffer(&ctx, login_buffer.data(), login_buffer.size());

    std::vector<uint8_t> true_vect{ 132,233,3,24,93,247,166,194,21,206,134,197,148,45,22,24 };



    if (true_vect != login_buffer) {
        passwordMatch = false;
    }

    if (!passwordMatch) {
        MessageBox(NULL, L":(", L"Noooo...", MB_OK | MB_ICONERROR);
        return;
    }

        unsigned long sum = 0;

    for (int i = 0; i < login_buffer.size(); i++) {
        sum += login_buffer[i];
    }

    for (int i = 0; i < login_buffer.size(); i++) {
        sum ^= login_buffer[i];
    }

    unsigned long temp = 0;

for (int i = 0; i < part4; i++) {
    part1 ^= part2;
    part3 += part1;
    part1 -= 1337;
    part2 ^= 0x33333;
    part3 -= 555;
    part3 ^= part1 + part2;

    part1 ^= sum;
    part2 ^= sum;
    part3 ^= sum;
```

```
    part1 ^= 5558;
    part2 += 5446546;
    part3 -= 'Kuba';
}



if (part1 != 3820249944 || part2 != 1941409244 || part3 != 2563851696 || part4 != 0x1337)
    passwordMatch = false;

    if (passwordMatch) {
        MessageBox(NULL, L"It's a flag! Congratulations! :)", L"FLAG!!!", MB_OK);
    }
    else
        MessageBox(NULL, L":(", L"Noooo...", MB_OK | MB_ICONERROR);
```

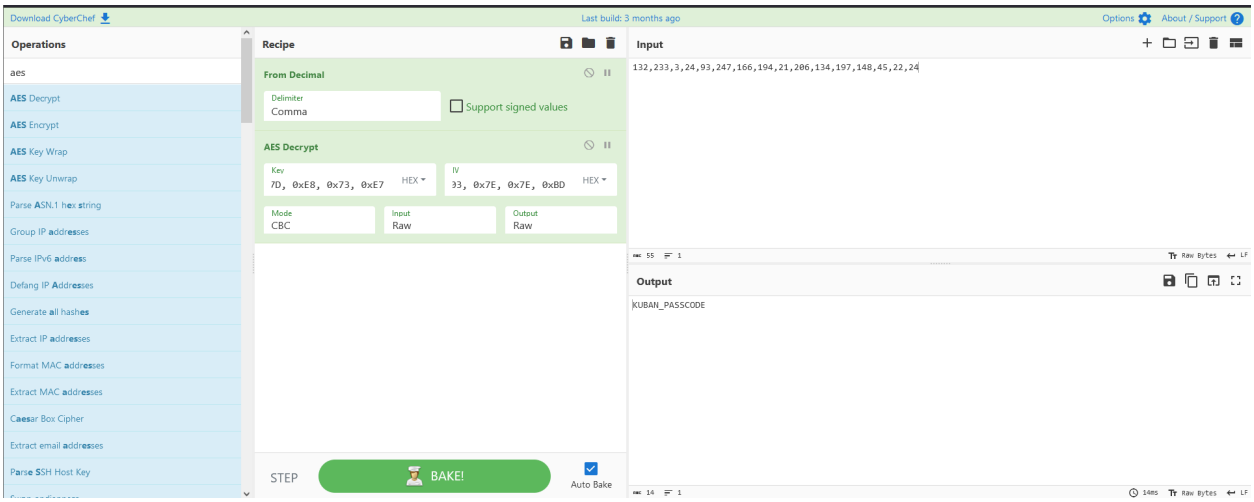
Тут используется AES для кодовой фразы и побитовое шифрование для флага.

Используя IV и ключ, можно расшифровать фразу-код из вектора:

```
132,233,3,24,93,247,166,194,21,206,134,197,148,45,22,24
```

Для кибершефа:

#recipe=From_Decimal('Comma',false)AES_Decrypt(%7B'option':'Hex','string':'0xD6,%200x23,%200xB8,%2



Пароль-фраза: `KUBAN_PASSCODE`.

```
unsigned long sum = 0;

for (int i = 0; i < login_buffer.size(); i++) {
    sum += login_buffer[i];
}
```

```
for (int i = 0; i < login_buffer.size(); i++) {
    sum ^= login_buffer[i];
}
```

Теперь в ходе этого цикла `sum` всегда будет 2038.

Флаг будет формата: `CSC{XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX}`

Тогда цикл будет такой:

```
for (int i = 0; i < part4; i++) {
    part1 ^= part2;
    part3 += part1;
    part1 -= 1337;
    part2 ^= 0x33333;
    part3 -= 555;
    part3 ^= part1 + part2;

    part1 ^= sum;
    part2 ^= sum;
    part3 ^= sum;

    part1 ^= 5558;
    part2 += 5446546;
    part3 -= 'Kuba';
}



if (part1 != 3820249944 || part2 != 1941409244 || part3 != 2563851696 || part4 != 0x1337)
    passwordMatch = false;
```

`part4` всегда будет `00001337`.

Остальные части можно вычислить из ожидаемых значений.

```
#include <stdio.h>

int main() {

        unsigned long part1, part2, part3, part4, temp;

    part1 = 3820249944;
        part2 = 1941409244;
        part3 = 2563851696;
        part4 = 0x1337;
```

```
    for (int i = 0; i < part4; i++) {
        part3 += 'Kuba';
        part2 -= 5446546;
                part1 ^= 5558;

        part3 ^= 2038;
        part2 ^= 2038;
        part1 ^= 2038;

        part3 ^= part1 + part2;
        part3 += 555;
        part2 ^= 0x33333;
        part1 += 1337;
        part3 -= part1;
        part1 ^= part2;
    }

        printf("CSC{%X-%X-%X-0000%X}", part1, part2, part3, part4);
        return 0;
}
```