



Название:	Добро пожаловать
Категория:	Реверс-инжиниринг
Уровень:	Лёгкий
Очки:	300
Описание:	Добро пожаловать, разомнись перед боем
Автор:	ROP

Прохождение:

Изучаем файл в IDA.

```
mov    eax, 17h
mov    rcx, rax
call   magic
lea    rdx, [rbp+Buffer] ; Str2
lea    rax, [rbp+Str1]
mov    r8d, 17h          ; MaxCount
mov    rcx, rax          ; Str1
call   strncmp
test   eax, eax
inz    short loc 401674
```

```
mov    eax, 17h
mov    rcx, rax
call   magic
lea    rdx, [rbp+Buffer] ; Str2
lea    rax, [rbp+Str1]
mov    r8d, 17h          ; MaxCount
mov    rcx, rax          ; Str1
call   strncmp
test   eax, eax
inz    short loc 401674
```

ФУНКЦИЯ `magic` обрабатывает наш ввод. А точнее шифрует.

The screenshot shows the IDA Pseudocode-A view. The assembly code is as follows:

```
1 |__int64 __fastcall magic(__int64 a1, int a2)
2 |
3 |    __int64 result; // rax
4 |    unsigned int i; // [rsp+Ch] [rbp-4h]
5 |
6 |    for ( i = 0; ; ++i )
7 |    {
8 |        result = i;
9 |        if ( (int)i >= a2 )
10 |            break;
11 |        *(BYTE*)(a1 + (int)i) ^= 1u;
12 |        *(BYTE*)(a1 + (int)i) += 3;
13 |        *(BYTE*)(a1 + (int)i) ^= 3u;
14 |        *(BYTE*)(a1 + (int)i) -= 7;
15 |
16 |    return result;
17 }
```

The screenshot shows the IDA Pseudocode-A view, identical to the one above, displaying the same assembly code for the function `magic`.

```
1 |__int64 __fastcall magic(__int64 a1, int a2)
2 |
3 |    __int64 result; // rax
4 |    unsigned int i; // [rsp+Ch] [rbp-4h]
5 |
6 |    for ( i = 0; ; ++i )
7 |    {
8 |        result = i;
9 |        if ( (int)i >= a2 )
10 |            break;
11 |        *(BYTE*)(a1 + (int)i) ^= 1u;
12 |        *(BYTE*)(a1 + (int)i) += 3;
13 |        *(BYTE*)(a1 + (int)i) ^= 3u;
14 |        *(BYTE*)(a1 + (int)i) -= 7;
15 |
16 |    return result;
17 }
```

Затем сравнение в `strcmp` с этим массивом:

```
mov    [rbp+Str1], 3Fh ; '?'
mov    [rbp+var_4F], 4Fh ; 'O'
mov    [rbp+var_4E], 3Fh ; '?'
mov    [rbp+var_4D], 77h ; 'w'
mov    [rbp+var_4C], 5Dh ; ']'
mov    [rbp+var_4B], 39h ; '9'
mov    [rbp+var_4A], 6Fh ; 'o'
mov    [rbp+var_49], 51h ; 'Q'
mov    [rbp+var_48], 5Bh ; '['
mov    [rbp+var_47], 6Eh ; 'n'
mov    [rbp+var_46], 2Eh ; '.'
mov    [rbp+var_45], 72h ; 'r'
mov    [rbp+var_44], 3Dh ; '='
mov    [rbp+var_43], 4Eh ; 'N'
mov    [rbp+var_42], 6Fh ; 'o'
mov    [rbp+var_41], 5Dh ; ']'
mov    [rbp+var_40], 5Bh ; '['
mov    [rbp+var_3F], 5Eh ; '^'
mov    [rbp+var_3E], 2Eh ; '.'
mov    [rbp+var_3D], 64h ; 'd'
mov    [rbp+var_3C], 64h ; 'd'
mov    [rbp+var_3B], 71h ; 'q'
mov    [rbp+var_3A], 75h ; 'u'
lea    rcx, Format      ; "Hey, buddy, you said you owned a flag...."
call   printf
```

```
mov    [rbp+Str1], 3Fh ; '?'
mov    [rbp+var_4F], 4Fh ; 'O'
mov    [rbp+var_4E], 3Fh ; '?'
mov    [rbp+var_4D], 77h ; 'w'
mov    [rbp+var_4C], 5Dh ; ']'
mov    [rbp+var_4B], 39h ; '9'
mov    [rbp+var_4A], 6Fh ; 'o'
mov    [rbp+var_49], 51h ; 'Q'
mov    [rbp+var_48], 5Bh ; '['
mov    [rbp+var_47], 6Eh ; 'n'
mov    [rbp+var_46], 2Eh ; '.'
mov    [rbp+var_45], 72h ; 'r'
mov    [rbp+var_44], 3Dh ; '='
mov    [rbp+var_43], 4Eh ; 'N'
mov    [rbp+var_42], 6Fh ; 'o'
mov    [rbp+var_41], 5Dh ; ']'
mov    [rbp+var_40], 5Bh ; '['
mov    [rbp+var_3F], 5Eh ; '^'
mov    [rbp+var_3E], 2Eh ; '.'
mov    [rbp+var_3D], 64h ; 'd'
mov    [rbp+var_3C], 64h ; 'd'
mov    [rbp+var_3B], 71h ; 'q'
mov    [rbp+var_3A], 75h ; 'u'
lea    rcx, Format      ; "Hey, buddy, you said you owned a flag...."
call   printf
```

Дешифруем этот массив:

[https://cyberchef.org/#recipe=ADD\({'option':'Hex','string':'7'}\).XOR\({'option':'Hex','string':'3'}\)](https://cyberchef.org/#recipe=ADD({'option':'Hex','string':'7'}).XOR({'option':'Hex','string':'3'}))