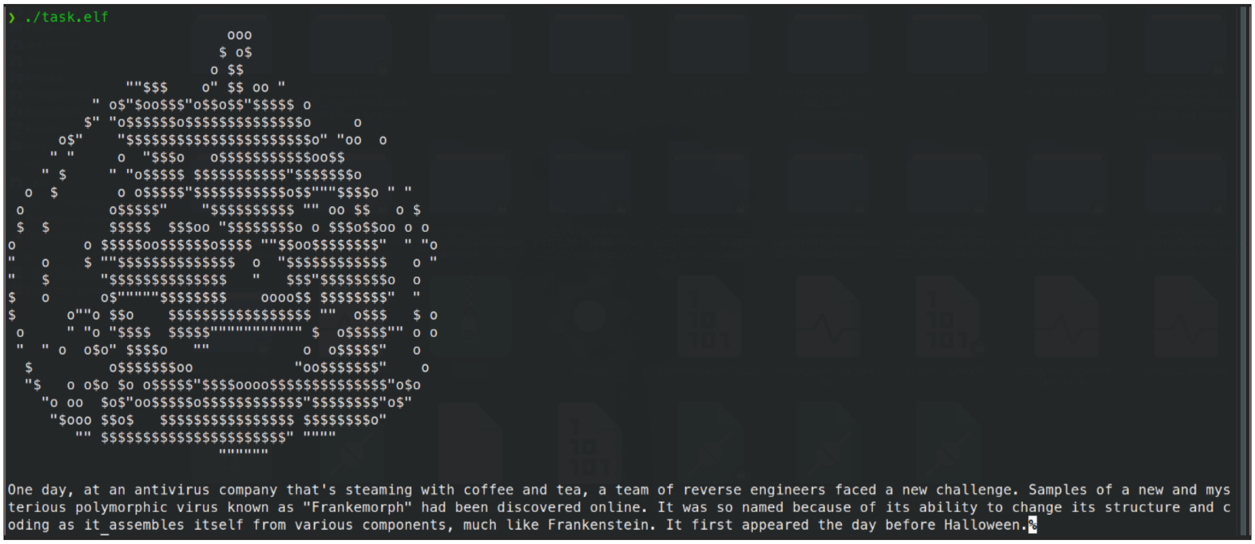
Название:	Тыква
Категория:	Реверс-инжиниринг
Уровень:	Легко
Очки:	350
Описание:	Это интересная история! Только вот конец...
Теги:	C
Автор:	ROP

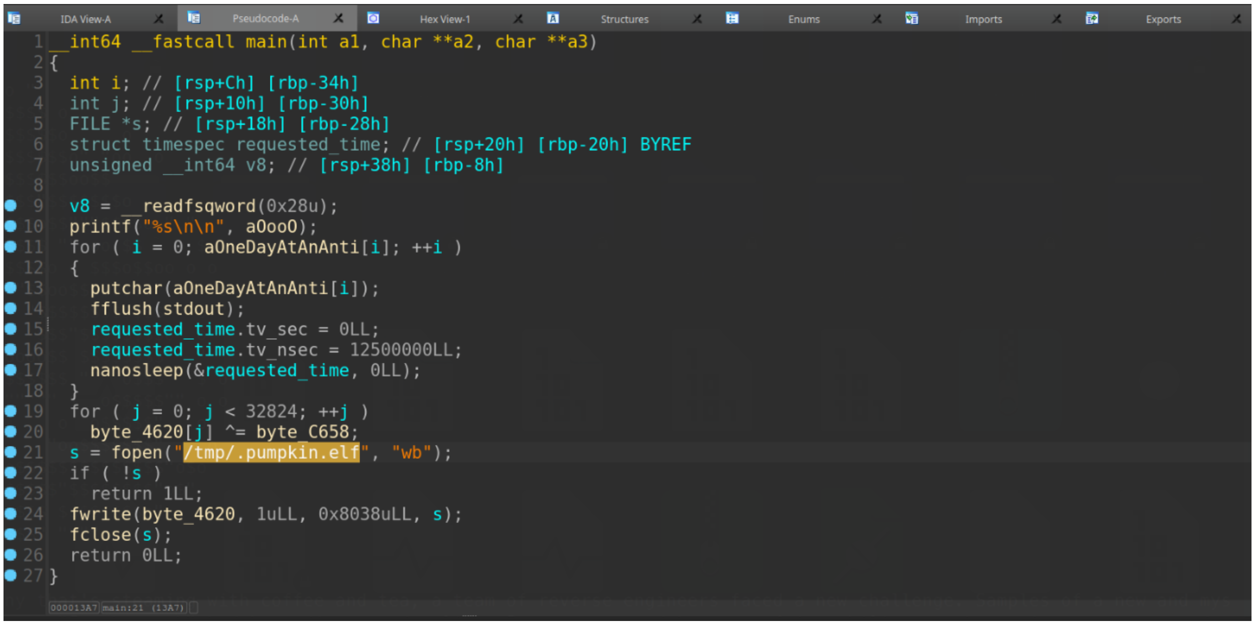
Прохождение:

Распаковываем архив, запускаем.



```
❯ ./task.elf
                          ooo
                         $ o$
                        o $$
              ""$$$    o" $$ oo "
          " o$"$oo$$$"o$$o$$"$$$$$ o
         $" "o$$$$$$o$$$$$$$$$$$$$$o     o
      o$"    "$$$$$$$$$$$$$$$$$$$$$$o" "oo  o
     " "     o  "$$$o   o$$$$$$$$$$$oo$$
    " $     " "o$$$$$ $$$$$$$$$$$"$$$$$$$o
  o  $       o o$$$$$"$$$$$$$$$$$o$$"""$$$$o " "
 o          o$$$$$"    "$$$$$$$$$$ "" oo $$   o $
 $  $       $$$$$  $$$oo "$$$$$$$$o o $$$o$$oo o o
o        o $$$$$oo$$$$$$o$$$$ ""$$oo$$$$$$$$"  " "o
"   o    $ ""$$$$$$$$$$$$$$  o  "$$$$$$$$$$$$   o "
"   $      "$$$$$$$$$$$$$$   "   $$$"$$$$$$$$o  o
$   o      o$"""""$$$$$$$$    oooo$$ $$$$$$$$"  "
$      o""o $$o    $$$$$$$$$$$$$$$$$ ""  o$$$   $ o
 o     " "o "$$$$  $$$$$""""""""""" $  o$$$$$"" o o
 "  " o  o$o" $$$$o   ""           o  o$$$$$"   o
  $         o$$$$$$$$oo           "oo$$$$$$$"    o
  "$   o o$o $o o$$$$$"$$$$oooo$$$$$$$$$$$$$$"o$o
    "o oo  $o$"oo$$$$$o$$$$$$$$$$$$"$$$$$$$$"o$"
     "$ooo $$o$   $$$$$$$$$$$$$$$$ $$$$$$$$o"
        "" $$$$$$$$$$$$$$$$$$$$$$" """"
                         """"""

One day, at an antivirus company that's steaming with coffee and tea, a team of reverse engineers faced a new challenge. Samples of a new and mys
terious polymorphic virus known as "Frankemorph" had been discovered online. It was so named because of its ability to change its structure and c
oding as it_assembles itself from various components, much like Frankenstein. It first appeared the day before Halloween.%
```

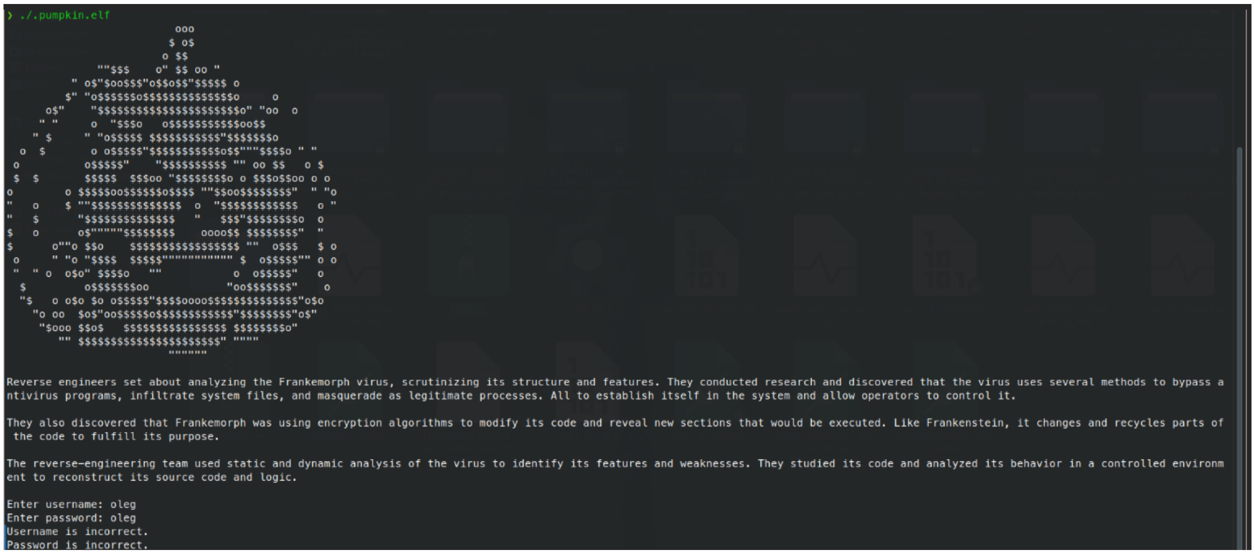
История от тыквы. Идём в IDA.



```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
  int i; // [rsp+Ch] [rbp-34h]
  int j; // [rsp+10h] [rbp-30h]
  FILE *s; // [rsp+18h] [rbp-28h]
  struct timespec requested_time; // [rsp+20h] [rbp-20h] BYREF
  unsigned __int64 v8; // [rsp+38h] [rbp-8h]

  v8 = __readfsqword(0x28u);
  printf("%s\n\n", a0oo0);
  for ( i = 0; aOneDayAtAnAnti[i]; ++i )
  {
    putchar(aOneDayAtAnAnti[i]);
    fflush(stdout);
    requested_time.tv_sec = 0LL;
    requested_time.tv_nsec = 12500000LL;
    nanosleep(&requested_time, 0LL);
  }
  for ( j = 0; j < 32824; ++j )
    byte_4620[j] ^= byte_C658;
  s = fopen("/tmp/.pumpkin.elf", "wb");
  if ( !s )
    return 1LL;
  fwrite(byte_4620, 1uLL, 0x8038uLL, s);
  fclose(s);
  return 0LL;
}
```

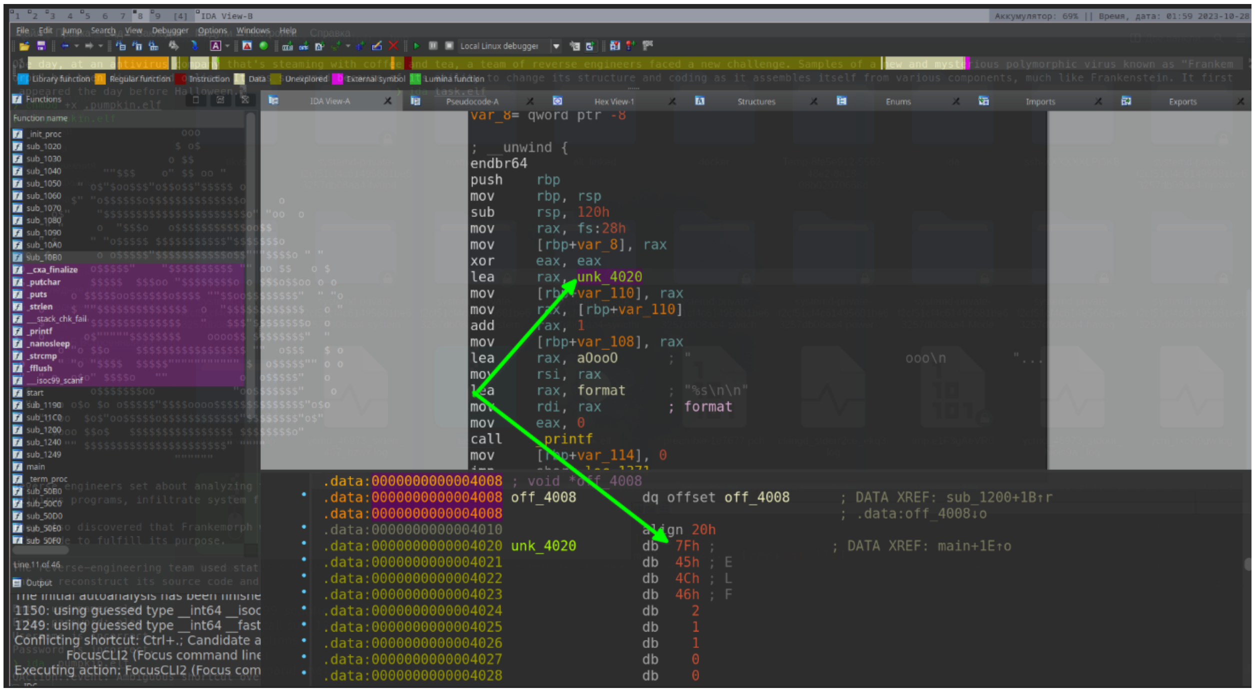
По такому пути пишется ELF-файл. Проверяем.

Да, всё на месте.



```
❯ ./.pumpkin.elf
                          ooo
                         $ o$
                        o $$
              ""$$$    o" $$ oo "
          " o$"$oo$$$"o$$o$$"$$$$$ o
         $" "o$$$$$$o$$$$$$$$$$$$$$o     o
      o$"    "$$$$$$$$$$$$$$$$$$$$$$o" "oo  o
     " "     o  "$$$o   o$$$$$$$$$$$oo$$
    " $     " "o$$$$$ $$$$$$$$$$$"$$$$$$$o
  o  $       o o$$$$$"$$$$$$$$$$$o$$"""$$$$o " "
 o          o$$$$$"    "$$$$$$$$$$ "" oo $$   o $
 $  $       $$$$$  $$$oo "$$$$$$$$o o $$$o$$oo o o
o        o $$$$$oo$$$$$$o$$$$ ""$$oo$$$$$$$$"  " "o
"   o    $ ""$$$$$$$$$$$$$$  o  "$$$$$$$$$$$$   o "
"   $      "$$$$$$$$$$$$$$   "   $$$"$$$$$$$$o  o
$   o      o$"""""$$$$$$$$    oooo$$ $$$$$$$$"  "
$      o""o $$o    $$$$$$$$$$$$$$$$$ ""  o$$$   $ o
 o     " "o "$$$$  $$$$$""""""""""" $  o$$$$$"" o o
 "  " o  o$o" $$$$o   ""            o  o$$$$$"   o
  $         o$$$$$$$$oo           "oo$$$$$$$"    o
  "$    o o$o $o o$$$$$"$$$$oooo$$$$$$$$$$$$$$"o$o
    "o oo  $o$"oo$$$$$o$$$$$$$$$$$$"$$$$$$$$"o$"
     "$ooo $$o$   $$$$$$$$$$$$$$$$ $$$$$$$$o"
        "" $$$$$$$$$$$$$$$$$$$$$$$" """"
                        """"""

Reverse engineers set about analyzing the Frankemorph virus, scrutinizing its structure and features. They conducted research and discovered that the virus uses several methods to bypass a
ntivirus programs, infiltrate system files, and masquerade as legitimate processes. All to establish itself in the system and allow operators to control it.

They also discovered that Frankemorph was using encryption algorithms to modify its code and reveal new sections that would be executed. Like Frankenstein, it changes and recycles parts of
 the code to fulfill its purpose.

The reverse-engineering team used static and dynamic analysis of the virus to identify its features and weaknesses. They studied its code and analyzed its behavior in a controlled environm
ent to reconstruct its source code and logic.

Enter username: oleg
Enter password: oleg
Username is incorrect.
Password is incorrect.
```

Откроем новый файл в IDA.



```
_init_proc
sub_1020
sub_1030
sub_1040
sub_1050
sub_1060
sub_1070
sub_1080
sub_1090
sub_10A0
sub_10B0
__cxa_finalize
_putchar
_puts
_strlen
___stack_chk_fail
_printf
_nanosleep
_strcmp
_fflush
___isoc99_scanf
start
sub_1190
sub_11C0
sub_1200
sub_1240
sub_1249
main
_term_proc
sub_50B0
sub_50C0
sub_50D0
sub_50E0
sub_50F0
```

Line 11 of 46

```
The initial autoanalysis has been finishe
1150: using guessed type __int64 __isoc
1249: using guessed type __int64 __fast
Conflicting shortcut: Ctrl+.; Candidate a
      FocusCLI2 (Focus command line
Executing action: FocusCLI2 (Focus com
```

```
var_8= qword ptr -8

; __unwind {
endbr64
push    rbp
mov     rbp, rsp
sub     rsp, 120h
mov     rax, fs:28h
mov     [rbp+var_8], rax
xor     eax, eax
lea     rax, unk_4020
mov     [rbp+var_110], rax
mov     rax, [rbp+var_110]
add     rax, 1
mov     [rbp+var_108], rax
lea     rax, aOooO      ; "                    ooo\n          "...
mov     rsi, rax
lea     rax, format     ; "%s\n\n"
mov     rdi, rax        ; format
mov     eax, 0
call    _printf
mov     [rbp+var_114], 0
```

```
.data:0000000000004008 ; void *off_4008
.data:0000000000004008 off_4008        dq offset off_4008      ; DATA XREF: sub_1200+1B↑r
.data:0000000000004008                                         ; .data:off_4008↓o
.data:0000000000004010                 align 20h
.data:0000000000004020 unk_4020        db  7Fh ;               ; DATA XREF: main+1E↑o
.data:0000000000004021                 db  45h ; E
.data:0000000000004022                 db  4Ch ; L
.data:0000000000004023                 db  46h ; F
.data:0000000000004024                 db    2
.data:0000000000004025                 db    1
.data:0000000000004026                 db    1
.data:0000000000004027                 db    0
.data:0000000000004028                 db    0
```

В памяти есть другой ELF-файл Попробуем его сдампить оттуда.

Сдампим от начала .data до конца этой секции. Тогда размер файла будет `0x87C1-0x4020 = 0x47A1`.

```
auto fname      = "/tmp/p3.elf";
auto address    = 0x4020;
auto size       = 0x47A1;
auto file= fopen(fname, "wb");

savefile(file, 0, address, size);
fclose(file);
```

Проверяем новый файл. Там будет флаг.