



Название:	По дороге к Замку Капибар
Категория:	Реверс-инжиниринг
Уровень:	Средний
Очки:	700
Описание:	На пути к флагу!
Теги:	C
Автор:	ROP

Прохождение:

Попробуем запустить файл.

```
> ./task
По дороге к Замку Капибар

Вы живете в маленькой деревушке, окруженной зелеными лесами и живописными полями. Все жители деревни говорят о загадочном Замке Капибар, который находится глубоко в лесу. Легенда гласит, что владелец замка, Король Капибар, является мудрым и щедрым правителем, и у него есть особый флаг, который приносит удачу тому, кто его обретет.
Один из жителей деревни, старый мудрец, рассказывает вам о своей встрече с Королем Капибар и предлагает отправиться в путешествие к Замку Капибар, чтобы попросить у Короля флаг. Вы восхищены этой идеей и решаете отправиться в это захватывающее приключение.
Вы отправляетесь из своей деревни на захватывающее приключение, чтобы добраться до замка Короля капибар и получить у него ценный флаг. Ваш путь начинается в заросших джунглях, где вы и ваши спутники готовы преодолеть любые трудности, чтобы достичь своей цели.

Задание 1: Мост разрушен
Прибыв в джунгли, вы обнаруживаете, что мост, который обычно ведет к Замку капибар, разрушен. Вам необходимо найти способ перебраться на другую сторону.

1. Искать путь вокруг разрушенного моста
2. Построить временный мост из имеющихся материалов

Выберите вариант (1 или 2): ■
```

Какой-то квест. Попробуем пройти.

Задание 2: Тропические преграды

Вам придется преодолеть несколько испытаний, чтобы продолжить свой путь.

1. Столкнуться с грозным тигром
2. Перебраться через глубокую реку
3. Пройти через море колючих зарослей

Выберите вариант (1, 2 или 3): 2

Вы перебрались через глубокую реку и продолжили свой путь.

Задание 3: Тайна замка

Приближаясь к замку, вы встречаете таинственную дверь, охраняемую заморскими стражниками. Они задают вам вопрос: "Какое количество лап у калибара?". Вам нужно ответить правильно, чтобы они пропустили вас внутрь.

1. 4
2. 1337

Выберите вариант (1 или 2): 1

Вы решаете головоломку и получаете ключ. Вы продолжаете свой путь.

Задание 4: Встреча с Королём калибар

Наконец, вы достигаете Замка Калибар и встречаетесь с Королём Калибаром. Он величественно сидит на троне, держа в руках флаг, который вы так долго искали. Он предлагает вам сыграть с ним в игру, чтобы получить флаг. Вам предстоит отгадывать загадки о природе и джунглях.

1. Ответить на вопросы о джунглях и природе, используя ваши чувства
2. Ответить на вопросы о джунглях и природе, используя ваши знания

Выберите вариант (1 или 2): 1

Вы ответили правильно на вопросы о джунглях и природе. Король калибар доверяет вам и предоставляет флаг. Вы почти завершаете квест со всеми почестями!

Король Калибар даёт вам флаг. Вам осталось только его прочитать. Что вы видите?

Что вы прочитали: asdf

Вы не смогли прочитать флаг()

ɔ ~ /Documents/CTF_megogr/3/task > █

1m 6s O 14:56:28

Чтение флага... Посмотрим в IDA.

The screenshot shows the IDA Pro interface with the following details:

- File menu:** File, Edit, Jump, Search, View, Debugger, Options, Windows, Help.
- Toolbar:** Includes icons for file operations, search, and debugger controls.
- Status bar:** Local Linux debugger, 100.00% (-151,169) (477,206) 000019C7 00000000000079C7: start+B (Synchronized with Hex View-1).
- Function list:** Shows numerous sub-functions, mostly named sub_4F5 through sub_65C5.
- Assembly view:** Displays the assembly code for the current function:

```
push    rdx
call    loc_7C30
push    rbp
push    rbx
push    rcx
push    rdx
add     rsi, rdi
push    rsi
mov    rsi, rdi
mov    rdi, rdx
xor    ebx, ebx
xor    ecx, ecx
or     rbp, 0xFFFFFFFFFFFFFFFh
call    sub_7A2A
add     ebx, ebx
jz     short loc_79E0
```
- Call stack:** A call graph window is visible on the left.
- Output window:** Contains the following text:

```
The decompilation hotkey is F5.
Please check the Edit/Plugins menu for more information.
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
```
- Bottom status:** IDC, ADS - idle, Down, Disk: 60GB.

A yellow box highlights the assembly code, and a yellow arrow points from the assembly code to a callout box containing the instruction `rep retn`. Another yellow arrow points from the assembly code to the `loc_79E0:` label. A third yellow arrow points from the assembly code to the `start endp ; sp-analysis failed` text in the output window.

ВОТ ЭТО ГОВОРИТ О ТОМ, ЧТО ФАЙЛ ЗАПАКОВАН ЧЕРЕЗ UPX.

```
> ./upx -d task
          Ultimate Packer for eXecutables
          Copyright (C) 1996 – 2023
UPX 4.0.2      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 30th 2023

  File size      Ratio      Format      Name
-----
upx: task: CantUnpackException: l_info corrupted

Unpacked 1 file: 0 ok, 1 error.
q ~/Documents/CTF_meropr/3/task >
```

Попробуем распаковать через upx -d task . Ошибка. Если игрок знает структуру UPX, то зайдёт в HEX-

редактор и поменяет подменённый magic UPX'а на UPX! .

БЫЛО .

Open task /home/rop/Documents/CTF_meropr/3/task

00000000	7F 45 4C 46 02 01 01 00 00 00 00 00 00 00 00 03 00 3E 00 01 00	.ELF.....>...
00000016	00 00 B8 79 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00	...y.....@.....
0000002C	00 00 00 00 00 00 00 00 40 00 38 00 03 00 00 00 00 00 00 00 00 01 00@.8.....
00000042	00 00 06 00PP.....
00000058	00 00 00 00 00 00 00 00 10 00 00 00 00 00 00 00 50 50 00 00 00 00 00 00 00
0000006E	00 00 00 10 00 00 00 00 00 00 01 00 00 00 05 00 00 00 00 00 00 00 00 00 00
00000084	00 00 00 00 00 60 00 00 00 00 00 00 00 60 00 00 00 00 00 00 00 56 24'.....VS\$
0000009A	00 00 00 00 00 56 24 00 00 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00VS\$.....
000000B0	51 E5 74 64 06 00	Q.td.....
000000C6	00 00
000000DC	00 00 00 00 10 00 00 00 00 00 00 C3 DC 72 04 43 44 42 21 A8 0Ax.CDB!..
000000F2	0E 16 00 00 00 00 A8 48 00 00 FC 1B 00 00 18 03 00 00 D3 00 00 00H.....
00000108	02 00 00 00 F6 FB 21 FF 7F 45 4C 46 02 01 01 00 03 00 3E 00 0D 80!..ELF.....>...
0000011E	11 0F 77 C9 0E 76 40 17 68 41 22 13 38 00 0D B2 65 DD 77 05 1D 00	..w..v@.hA".8...e.w...
00000134	1C 00 06 0F 04 27 07 2C D9 85 9C D8 02 08 67 37 85 BC B2 F3 18 03'.....g7.....
0000014A	07 1C 00 4E 61 6F C9 01 37 00 A8 08 96 5D F6 C2 07 00 10 37 05 0F	...Nao..7....]....7..
00000160	F6 16 72 CA 07 2D 0B 17 6F C9 2B 3B 13 20 07 FC 1B EE 6C 8E 6C 37	..r....o.+;..l.17
00000176	06 70 3D 07 42 9E BD 90 4D C4 02 37 E0 02 21 DF 19 E4 02 80 07 4D	.p=B...M..7...!.....M
0000018C	32 09 09 79 F0 01 4F 04 C8 29 98 0C 38 07 30 00 65 93 23 5B 37 68	2..y..0...)..8.0.e.#[7h
000001A2	07 FC 7B 0B 39 44 00 2B 6F 53 E5 74 64 9D 83 1D 90 50 37 A0 3A 43	..{.9D.+oS.td....P7.:C
000001B8	32 D8 95 07 4C 6F 4C 51 76 04 76 30 17 00 10 6F 76 20 E1 0C 52 87	2...LoLQv.v0...ov ..R.
000001CE	90 02 07 00 C0 30 16 67 01 00 00 00 00 00 20 FF 90 05 00 00 1E0.g.....
000001E4	02 00 00 02 00 00 DB 7F BB FD 2F 6C 69 62 36 34 05 64 2D 08 6E/lib64.d-n
000001FA	75 78 2D 78 38 36 2D 0F 2E 6F 9A EE FB 73 6F 2E 32 00 00 04 03 20	ux-x86-..o...so.2...
00000210	05 47 4E 55 00 64 1B 7B B6 02 13 C0 03 1D 00 0F 80 2F D8 17 E4 01	.GNU.d.{...../....
00000226	0B 14 1F 2F D8 F3 FF FF B6 EE 43 C0 0A 09 2B 70 24 46 2E 3C 00 25	.../.....C...+p\$F.<%
0000023C	11 89 4B AF 9C 0D 10 33 25 7B B3 2F 23 00 2F 5B 00 20 8F 0D F6 0B	.K...3%{./#./[.
00000252	0F 23 06 13 A1 00 80 13 D8 FB BF 3B D1 65 CE 6D 67 55 61 0A 00 35	.#.....:..e.mgUa..5
00000268	6C 36 80 74 17 12 B2 CF 2F 08 1B 42 A6 13 2E 17 86 90 21 64 18 67	16.t..../..B.....!d.g
0000027E	60 64 08 1B 42 01 A7 CE 42 86 B0 21 56 47 6E 21 64 08 19 DD 0D 00	'd..B...B..!VGn!d.....
00000294	69 86 B0 29 17 47 22 87 9D FC 6E 07 0F 11 00 1A 00 40 50 08 9A 66	i...).G"....n.....@P..f
000002AA	DB FD 97 FF 67 65 74 73 00 73 74 64 69 73 6C 65 65 70 00 70 75 10gets.stdSleep.pu.
000002C0	5F 5F 12 35 DD 6F 7F 61 63 6B 5F 63 68 03 66 61 69 6C 00 26 63 72	__5.o.ack_ch.fail.&cr
000002D6	DA BF 0E B7 22 6E 1C 8D 63 21 72 74 5F 6D 1D BB 6D E5 E6 11 28 78"n..c!rt_m..m...(x
000002EC	61 09 61 19 7A 65 27 BA DF DA 5B 2C 6C 6F 63 0B 09 6D 65 6D 63 6D	a.a.ze'...[,loc..memcm
00000302	53 72 B6 76 DB DD 1A 74 66 26 69 73 17 39 39 3A 1C 6E FD B7 8F D7	Srv...tf&is.99:.n....
00000318	0E 45 C2 36 B6 4C 49 42 43 5F 32 2E 37 49 B6 24 93 09 34 0B 35 AD	.E.6.LIBC_2.7I.\$..4.5.
0000032E	6D FF B7 33 16 5F 49 54 4D 5F 64 65 72 65 67 4B 74 06 0C 90 6F EE	m..3._ITM_deregKt...o.
00000344	B7 43 6A 6E 65 54 61 62 6D 98 67 6D 6F 6E 1A 12 D8 6C 5F 2A 28 D3	.CjneTabm.gmon...l_*(.
0000035A	02 6C 6C AC 6B 03 01 04 05 03 0D 05 13 18 5B 12 DE 01 F7 1F 7D CB	.ll.k.....[.....}.
00000370	0D F6 F4 2F 00 17 69 69 0D 65 87 13 14 3D 7D 90 EE 0F 23 91 75 1A	.../..ii.e...=)...#.u.
00000386	60 00 35 0D 7D B0 5F B2 0F B4 01 06 0E 6D A7 2E 7D 4D 0E 54 EC 8F

Offset: 0xEC

Стало.

task
/home/rop/Documents/CTF_mero/pr/3/task

```

00000000 7F 45 4C 46 02 01 01 00 00 00 00 00 00 00 00 00 00 03 00 3E 00 01 00 .ELF.....>...
00000016 00 00 B8 79 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 ..y.....@...
0000002C 00 00 00 00 00 00 00 00 40 00 38 00 03 00 00 00 00 00 00 00 00 00 01 00 .....@.8.....
00000042 00 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....PP.....
00000058 00 00 00 00 00 00 00 00 00 10 00 00 00 00 00 00 00 00 50 50 00 00 00 00 00 .....V$....
0000006E 00 00 00 10 00 00 00 00 00 00 01 00 00 00 05 00 00 00 00 00 00 00 00 00 00 .....V$....
00000084 00 00 00 00 60 00 00 00 00 00 00 00 60 00 00 00 00 00 00 00 00 00 56 24 .....V$....
0000009A 00 00 00 00 00 56 24 00 00 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 .....Q.td.....
000000B0 51 E5 74 64 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....UPX!.....
000000C6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....H.....
000000DC 00 00 00 10 00 00 00 00 00 00 C3 DC 72 04 55 50 58 21 A8 0A .....!..ELF.....
000000F2 0E 16 00 00 00 A8 48 00 00 FC 1B 00 00 18 03 00 00 D3 00 00 00 .....w..v@.hA".8...e.w...
00000108 02 00 00 00 F6 FB 21 FF 7F 45 4C 46 02 01 01 00 03 00 3E 00 00 0D 80 .....'.....g7.....
0000011E 11 0F 77 C9 0E 76 40 17 68 41 22 13 38 00 0D B2 65 DD 77 05 1D 00 .....'.9D.+oS.td.....
00000134 1C 00 06 0F 04 27 07 2C D9 85 9C D8 02 08 67 37 85 BC B2 F3 18 03 .....P7.:C
0000014A 07 1C 00 4E 61 6F C9 01 37 00 A8 08 96 5D F6 C2 07 00 10 37 05 0F .....Nao..7....].
00000160 F6 16 72 CA 07 2D 0B 17 6F C9 2B 3B 13 20 07 FC 1B EE 6C 8E 6C 37 .....r...-o.+;.1.17
00000176 06 70 3D 07 42 9E BD 90 4D C4 02 37 E0 02 21 DF 19 E4 02 80 07 4D .....p=B..M..7..!.M
0000018C 32 09 09 79 F0 01 4F 04 C8 29 9B 0C 38 07 30 00 65 93 23 5B 37 68 2..y..0..).8.0.e.#[7h
000001A2 07 FC 7B 0B 39 44 00 2B 6F 53 E5 74 64 9D 83 1D 90 50 37 A0 3A 43 ..{.9D.+oS.td.....
000001B8 32 D8 95 07 4C 6F 4C 51 76 04 76 30 17 00 10 6F 76 20 E1 0C 52 87 2...LoLQv.v0...ov ..R.
000001CE 90 02 07 00 C0 30 16 67 01 00 00 00 00 00 20 FF 90 05 00 00 1E .....0.g..... .
000001E4 02 00 00 02 00 00 00 DB 7F BB FD 2F 6C 69 62 36 34 05 64 2D 08 6E ...../lib64.d.-n
000001FA 75 78 2D 78 38 36 2D 0F 2E 6F 9A EE FB 73 6F 2E 32 00 00 04 03 20 ux-x86...o...so.2...
00000210 05 47 4E 55 00 64 1B 7B B6 02 13 C0 03 1D 00 0F 80 2F D8 17 E4 01 .GNU.d.{...../...
00000226 0B 14 1F 2F D8 F3 FF FF B6 EE 43 C0 0A 09 2B 70 24 46 2E 3C 00 25 .../.....C...+p$F.<.%.
0000023C 11 89 4B AF 9C 0D 10 33 25 7B B3 2F 23 00 2F 5B 00 20 8F 0D F6 0B ..K....3%{./#./[. ....
00000252 0F 23 06 13 A1 00 80 13 D8 FB BF 3B D1 65 CE 6D 67 55 61 0A 00 35 ..#. ....;e.mgUa..5
00000268 6C 36 80 74 17 12 B2 CF 2F 08 1B 42 A6 13 2E 17 86 90 21 64 18 67 16.t....B.....!d.g
0000027E 60 64 08 1B 42 01 A7 CE 42 86 B0 21 56 47 6E 21 64 08 19 DD 0D 00 'd..B...B..!VGn!d.....
00000294 69 86 B0 29 17 47 22 87 9D FC 6E 07 0F 11 00 1A 00 40 50 08 9A 66 i..).G"...n.....@P..f
000002AA DB FD 97 FF 67 65 74 73 00 73 74 64 69 73 6C 65 65 70 00 70 75 10 ....gets.stdSleep.pu.
000002C0 5F 5F 12 35 DD 6F 7F 61 63 6B 5F 63 68 03 66 61 69 6C 00 26 63 72 __.5.o.ack_ch.fail.&cr
000002D6 DA BF 0E B7 22 6E 1C 8D 63 21 72 74 5F 6D 1D BB 6D E5 E6 11 28 78 ...."n..c!rt_m..m...(x
000002EC 61 09 61 19 7A 65 27 BA DF DA 5B 2C 6C 6F 63 0B 09 6D 65 6D 63 6D a.a.ze'...[,loc..memcm
00000302 53 72 B6 76 DB DD 1A 74 66 26 69 73 17 39 39 3A 1C 6E FD B7 8F D7 Sr.v...tf&is.99:.n.....
00000318 0E 45 C2 36 B6 4C 49 42 43 5F 32 2E 37 49 B6 24 93 09 34 0B 35 AD ..E.6.LIBC_2.7I.$..4.5.
0000032E 6D FF B7 33 16 5F 49 54 4D 5F 64 65 72 65 67 4B 74 06 0C 90 6F EE m..3._ITM_deregKt...o.
00000344 B7 43 6A 6E 65 54 61 62 6D 98 67 6D 6F 6E 1A 12 D8 6C 5F 2A 28 D3 .CjneTabm.gmon...l_*(. .
0000035A 02 6C 6C AC 6B 03 01 04 05 03 0D 05 13 18 5B 12 DE 01 F7 1F 7D CB ll.k.....[.....}.
00000370 0D F6 F4 2F 00 17 69 69 0D 65 87 13 14 3D 7D 90 EE 0F 23 91 75 1A .../.ii.e...=}...#.u.
00000380 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....'.
Offset: 0xEF

```

Теперь можно распаковать.

```
> ./upx -d task
          Ultimate Packer for eXecutables
          Copyright (C) 1996 - 2023
UPX 4.0.2      Markus Oberhumer, Laszlo Molnar & John Reiser  Jan 30th 2023

  File size       Ratio     Format      Name
-----
  22087 <-    10060    45.55%  linux/amd64  task

Unpacked 1 file.
q ~/Documents/CTF_meropr/3/task >
```

И уже в IDA можно увидеть исходный код.

```

loc_1A6A:
    mov    eax, [rbp+var_C]
    cmp    eax, 0
    jne    short loc_1A0B

loc_1A01:
    mov    esi, 1
    lea    rax, unk_34C8
    mov    rdi, rax
    call   sub_1576
    mov    edi, rax           ; seconds
    call   sleep
    lea    rax, byte_3600
    mov    rdi, rax           ; s
    call   puts
    lea    rax, byte_3760
    mov    rdi, rax           ; format
    mov    eax, 0
    call   _printf
    mov    rdi, stdin          ; stream
    call   _getc
    mov    eax, 0
    call   sub_179C           ; Функция проверки и чтения.
    mov    [rbp+var_D], al
    cmp    [rbp+var_D], 0
    jz    loc_1A00

loc_1A00:
    jmp   short loc_1A0B

loc_1A0B:
    mov    eax, [rbp+var_C]
    cmp    eax, 0
    jne    short loc_1A0B

loc_1A01:
    mov    esi, 1
    lea    rax, unk_34C8
    mov    rdi, rax
    call   sub_1576
    mov    edi, rax           ; seconds
    call   sleep
    lea    rax, byte_3600
    mov    rdi, rax           ; s
    call   puts
    lea    rax, byte_3760
    mov    rdi, rax           ; format
    mov    eax, 0
    call   _printf
    mov    rdi, stdin          ; stream
    call   _getc
    mov    eax, 0
    call   sub_179C           ; Функция проверки и чтения.
    mov    [rbp+var_D], al
    cmp    [rbp+var_D], 0
    jz    loc_1A00
    jmp   short loc_1A0B

```

Зайдём в функцию.

```
mov    eax, [rbp+var_54]
movsxd rdx, eax      ; n          BE 22 00
lea    rax, [rbp+s]
lea    rcx, unk_5010  ; Зашифрованные правильные байты.
mov    rsi, rcx      ; s2
mov    rdi, rax      ; s1
call   _memcmpr
test   eax, eax
jnz    short loc_155B

loc_12EA:
mov    eax, [rbp+var_58]
cdqe
movzx eax, [rbp+rax+s]
lea    edx, [rax-16h]
mov    eax, [rbp+var_58]
cdqe
mov    [rbp+rax+s], dl
mov    eax, [rbp+var_58]
cdqe
movzx eax, [rbp+rax+s]
lea    edx, [rax+6Eh]
mov    eax, [rbp+var_58]
cdqe
mov    [rbp+rax+s], dl
mov    eax, [rbp+var_58]
cdqe
movzx eax, [rbp+rax+s]
xor   eax, 0FFFFFFFA0h
mov    edx, eax
mov    eax, [rbp+var_58]
cdqe
mov    [rbp+rax+s], dl
mov    eax, [rbp+var_58]
cdqe
movzx eax, [rbp+rax+s]
xor   eax, 5Ch
mov    edx, eax
mov    eax, [rbp+var_58]
cdqe
mov    [rbp+rax+s], dl
mov    eax, [rbp+var_58]
cdqe
movzx eax, [rbp+rax+s]
xor   eax, 62h
mov    edx, eax
mov    eax, [rbp+var_58]
cdqe
mov    [rbp+rax+s], dl
mov    eax, [rbp+var_58]
cdqe
movzx eax, [rbp+rax+s]
lea    edx, [rax-71h]
mov    eax, [rbp+var_58]
cdqe
mov    [rbp+rax+s], dl
mov    eax, [rbp+var_58]
cdqe
movzx eax, [rbp+rax+s]
lea    edx, [rax-14h]
mov    eax, [rbp+var_58]
cdqe
mov    [rbp+rax+s], dl
mov    eax, [rbp+var_58]
cdqe
movzx eax, [rbp+rax+s]
lea    edx, [rax+15h]
mov    eax, [rbp+var_58]
cdqe
```

Видим алгоритм проверки и зашифрованные правильные байты
(`unk_5010`) .

```
• .data:0000000000005010 unk_5010      db  92h
  .data:0000000000005011                  db   6
  .data:0000000000005012                  db 0E5h
  .data:0000000000005013      010 92 06 E5 F0 db 0FCh 4A 92 F9
  .data:0000000000005014      020 05 4B 90 00 db 0FBh A2 F6 92
  .data:0000000000005015      030 00 DB B0 00 db 0F8h 00 00 27
  .data:0000000000005016      040 00 00 00 00 db 4Ah ; J
  .data:0000000000005017      050 00 00 00 00 db 92h 00 00 00
  .data:0000000000005018      060 00 00 00 00 db 0F9h 00 00 00
  .data:0000000000005019      070 00 00 00 00 db 49h ; I
  .data:000000000000501A      080 00 00 00 00 db 0F8h 00 00 00
  • .data:000000000000501B      090 00 00 00 00 db 5Bh ; [ 0 00
  .data:000000000000501C      0A0 00 00 00 00 db 0F9h 00 00 00
  .data:000000000000501D      0B0 00 00 00 00 db 4Bh ; K 00
  .data:000000000000501E      0C0 00 00 00 00 db 0A0h 00 00 00
  .data:000000000000501F                  db 0F6h
  .data:0000000000005020                  db 0D5h
  .data:0000000000005021                  db 4Bh ; K
  .data:0000000000005022                  db 90h
  .data:0000000000005023                  db 49h ; I
  .data:0000000000005024                  db 0F6h
  • .data:0000000000005025                  db 90h
  • .data:0000000000005026                  db 0A2h
  • .data:0000000000005027                  db 0F6h
  • .data:0000000000005028                  db 92h
  • .data:0000000000005029                  db 89h
  • .data:000000000000502A                  db 89h
  • .data:000000000000502B                  db 6Dh ; m
  • .data:000000000000502C                  db 54h ; T
  • .data:000000000000502D                  db 43h ; C
  • .data:000000000000502E                  db 0
  • .data:000000000000502F                  db 0
  • .data:0000000000005030 or more information db 0CDh
  • .data:0000000000005031                  db 0DBh
  • .data:0000000000005032 en propagated db 0BCh
  • .data:0000000000005033                  db 0CDh
  • .data:0000000000005033 _data           ends
```

Напишем дешифратор по алгоритму выше.

```
#include <stdio.h>

unsigned char check_pass[] = {
    146,
    6,
    229,
    252,
    251,
    248,
    74,
    146,
    249,
    73,
    248,
    91,
    249,
    75,
    160,
    246,
    213,
    75,
    144,
    73,
    246,
    144,
    162,
    246,
    146,
    137,
    137,
    109,
    84,
    67,
    0,
    0,
```

```
205,
219,
188,
205
};

int main() {

    int len_check = sizeof(check_pass) / sizeof(check_pass[0]);

    unsigned char i, j;

    char flag[50] = {0};

    for (j = 0; j < len_check; j++) {
        for (i = 0; i < 0x80; i++) {
            unsigned char temp = i;
            i -= 22;
            i += 110;
            i ^= 160;
            i ^= 92;
            i ^= 98;
            i += 143;
            i -= 20;
            i += 21;
            i -= 73;
            i ^= 76;
            i ^= 151;
            i ^= 56;
            i += 102;
            i -= 203;
            i += 86;
            i += 157;
            i -= 163;
            i += 198;
            i ^= 233;
```

```
i -= 147;
i += 28;
i ^= 212;
i -= 159;
i ^= 252;
i ^= 73;

if (i == check_pass[j])
{
    flag[j] = temp;
    break;
}

i = temp;
}
}

printf("%s\n", flag);
}
```

Запускаем и получаем флаг !