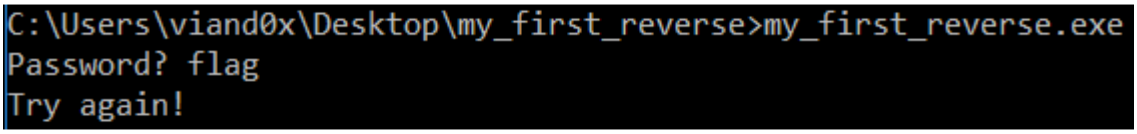
RITM ROOT IN THE MIDDLE

HOME RESSOURCES ABOUT

POSTED ON JANUARY 13, 2017 BY VIAND0X

Points: 50
Author: Frisk0
Description: Find the flag!

First thing to do : download and run the file provided during the challenge.

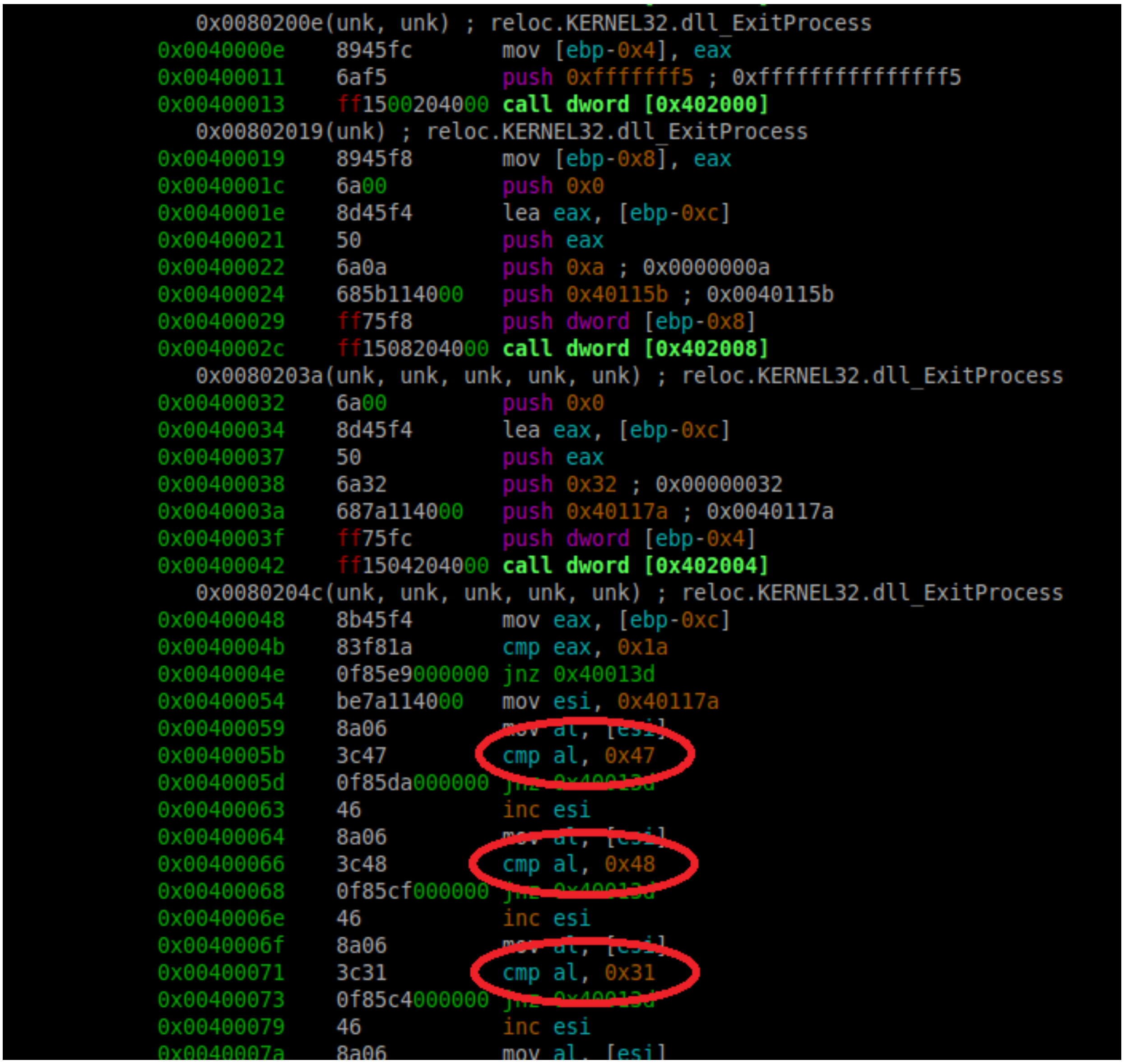


```
C:\Users\viand0x\Desktop\my_first_reverse>my_first_reverse.exe
Password? flag
Try again!
```

When we input something (in this case the word “flag”), the program outputs “Try again”.

Ok, so user’s input is compared to the flag during the process and the program indicates either the input corresponds to the flag or not.

So, let’s use **radare2** to see what’s going on.



```
   0x0080200e(unk, unk) ; reloc.KERNEL32.dll_ExitProcess
0x0040000e    8945fc       mov [ebp-0x4], eax
0x00400011    6af5         push 0xfffffff5 ; 0xfffffffffffffff5
0x00400013    ff1500204000 call dword [0x402000]
   0x00802019(unk) ; reloc.KERNEL32.dll_ExitProcess
0x00400019    8945f8       mov [ebp-0x8], eax
0x0040001c    6a00         push 0x0
0x0040001e    8d45f4       lea eax, [ebp-0xc]
0x00400021    50           push eax
0x00400022    6a0a         push 0xa ; 0x0000000a
0x00400024    685b114000   push 0x40115b ; 0x0040115b
0x00400029    ff75f8       push dword [ebp-0x8]
0x0040002c    ff1508204000 call dword [0x402008]
   0x0080203a(unk, unk, unk, unk, unk) ; reloc.KERNEL32.dll_ExitProcess
0x00400032    6a00         push 0x0
0x00400034    8d45f4       lea eax, [ebp-0xc]
0x00400037    50           push eax
0x00400038    6a32         push 0x32 ; 0x00000032
0x0040003a    687a114000   push 0x40117a ; 0x0040117a
0x0040003f    ff75fc       push dword [ebp-0x4]
0x00400042    ff1504204000 call dword [0x402004]
   0x0080204c(unk, unk, unk, unk, unk) ; reloc.KERNEL32.dll_ExitProcess
0x00400048    8b45f4       mov eax, [ebp-0xc]
0x0040004b    83f81a       cmp eax, 0x1a
0x0040004e    0f85e9000000 jnz 0x40013d
0x00400054    be7a114000   mov esi, 0x40117a
0x00400059    8a06         mov al, [esi]
0x0040005b    3c47         cmp al, 0x47
0x0040005d    0f85da000000 jnz 0x40013d
0x00400063    46           inc esi
0x00400064    8a06         mov al, [esi]
0x00400066    3c48         cmp al, 0x48
0x00400068    0f85cf000000 jnz 0x40013d
0x0040006e    46           inc esi
0x0040006f    8a06         mov al, [esi]
0x00400071    3c31         cmp al, 0x31
0x00400073    0f85c4000000 jnz 0x40013d
0x00400079    46           inc esi
0x0040007a    8a06         mov al, [esi]
```

As we can see, there are some char comparisons of the user input.

```
- jnz is a conditional jump to an adress (if the previous char
comparison fails, jump to 0x40013d)
- esi is incremented -> it points to next address (inc ASM instruction)
- al is re affected with current value pointed by esi, then compared
with another hex value
- this goes on until all comparisons are done
```

We can deduce that the flag is composed of all hex values our input is compared with . So Let's gather them all :

- **47 48 31 36 7b 72 33 76 33 72 73 31 6e 67 5f 31 73 5f 63 30 30 6c 21 7d**

Seems like the flag in ascii values ! Let's use any hex to ascii online conversion tool 🙂

```
GH16{r3v3rs1ng_1s_c00l!}
```

POSTED IN CTF, GREHACK2016

TAGGED 50, CRACKING, CTF, FIRST, GREHACK, MY, REVERSE, WRITEUP

LEAVE A REPLY

COMMENT

NAME *

EMAIL *

WEBSITE

POST COMMENT

This site uses Akismet to reduce spam. Learn how your comment data is processed.

Search ...

SEARCH

RECENT POSTS

Comment devenir root en 1 ligne ...

Write-up – Steganography 100 – Fucking flute

Write-up – Steganography 50 – Newbie challenge

NDH XV 2017 – RITM

RITM ROOT IN THE MIDDLE

HOME RESSOURCES ABOUT

Grehack2017	Insomni'hack2017
NDHXV	Securité
Tools	Tutorials

Copyright © 2018 rootinthemiddle.org – OnePress theme by FameThemes