

Category:

rev

Name:

ReverseMe

Message:

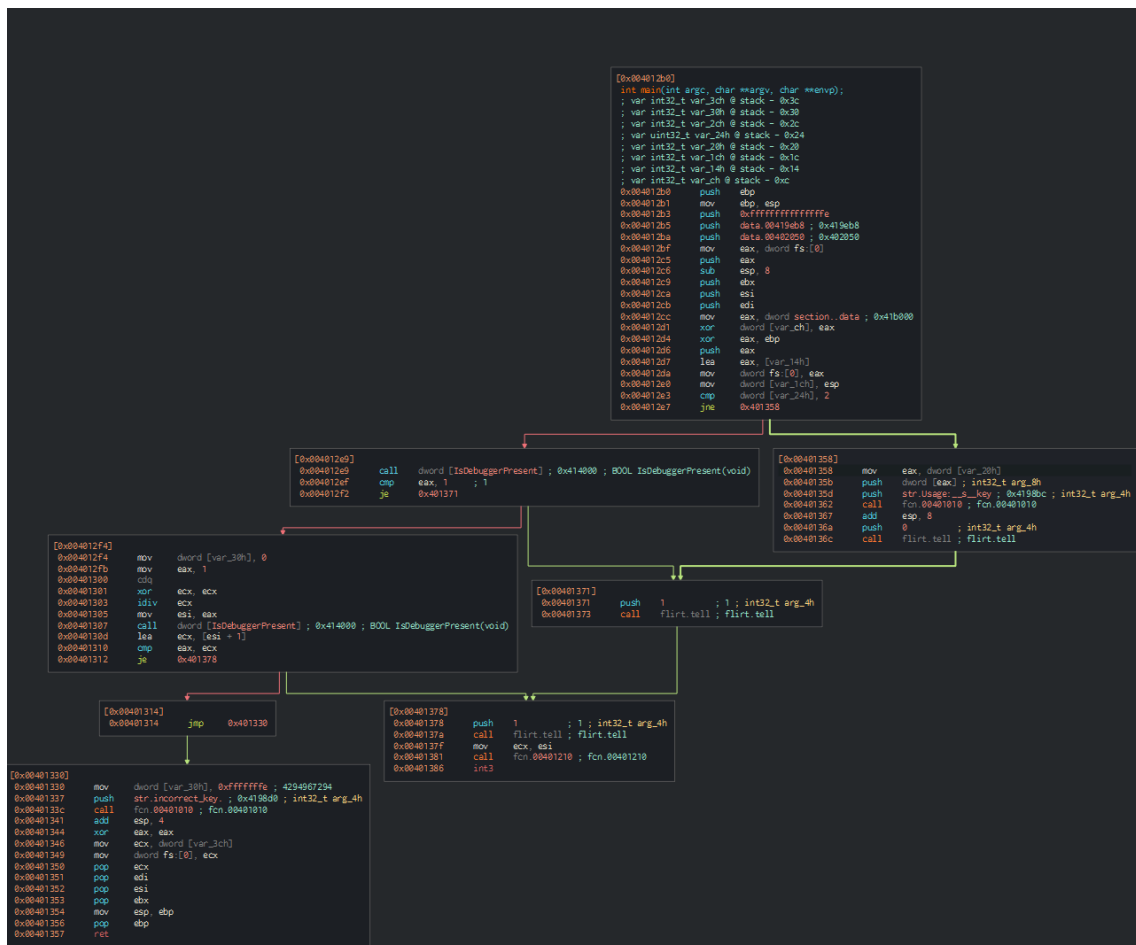
Reverse the file and get flag.

Instructions:

When run the file, the program shows the following message. Player needs to be identified the key.

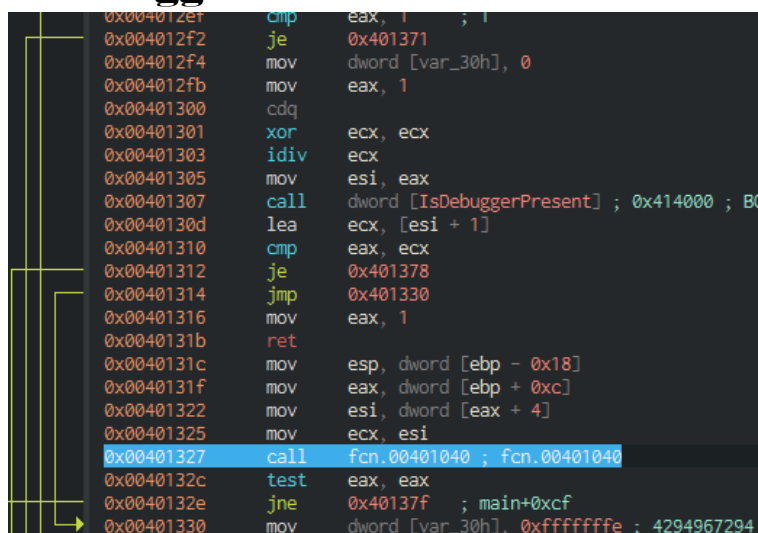
```
λ rev_chall.exe
Usage: rev_chall.exe <key>
```

Analyze the files. This document displays a screen example using Cutter.



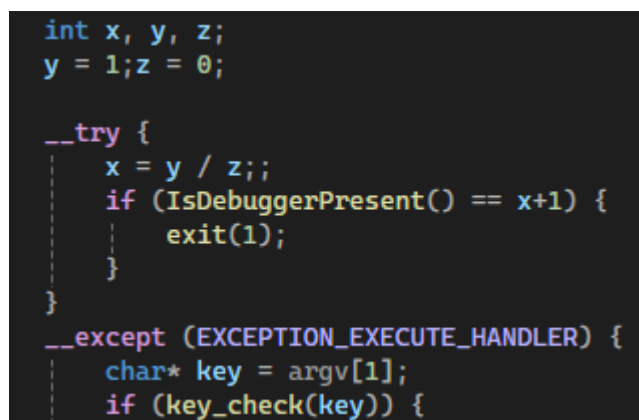
If you check the main function in the graph view, you will see

that it checks for the presence of arguments and "IsDebuggerPresent". However, no processing for handling argument values appears. The desired code is called in exception handling when a divide-by-zero error occurs. Normally, it would be difficult to perform dynamic analysis in a debugger.



```
0x004012ef  cmp     eax, 1 ; 1
0x004012f2  je      0x401371
0x004012f4  mov     dword [var_30h], 0
0x004012fb  mov     eax, 1
0x00401300  cdq
0x00401301  xor     ecx, ecx
0x00401303  idiv   ecx
0x00401305  mov     esi, eax
0x00401307  call   dword [IsDebuggerPresent] ; 0x414000 ; B0
0x0040130d  lea    ecx, [esi + 1]
0x00401310  cmp     eax, ecx
0x00401312  je      0x401378
0x00401314  jmp    0x401330
0x00401316  mov     eax, 1
0x0040131b  ret
0x0040131c  mov     esp, dword [ebp - 0x18]
0x0040131f  mov     eax, dword [ebp + 0xc]
0x00401322  mov     esi, dword [eax + 4]
0x00401325  mov     ecx, esi
0x00401327  call   fcn.00401040 ; fcn.00401040
0x0040132c  test   eax, eax
0x0040132e  jne    0x40137f ; main+0xcf
0x00401330  mov     dword [var_30h], 0xffffffff ; 4294967294
```

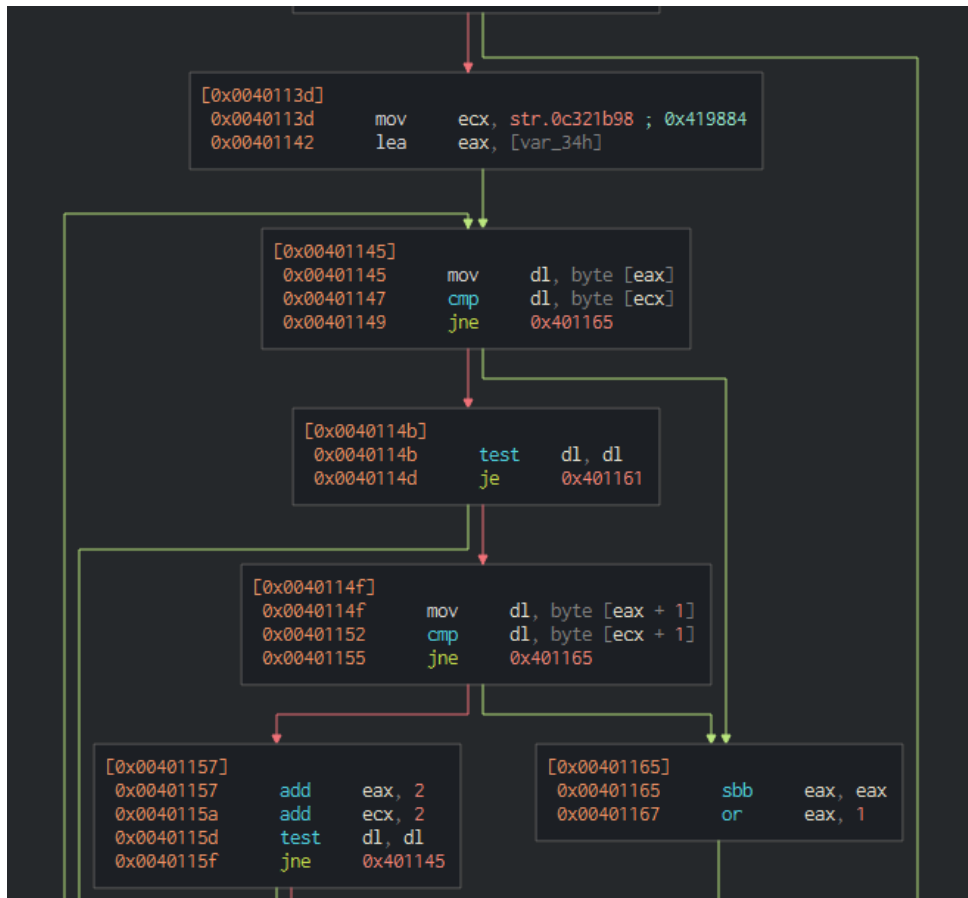
The source code is not distributed, but the relevant sections are compiled from the following code.



```
int x, y, z;
y = 1; z = 0;

__try {
    x = y / z;;
    if (IsDebuggerPresent() == x+1) {
        exit(1);
    }
}
__except (EXCEPTION_EXECUTE_HANDLER) {
    char* key = argv[1];
    if (key_check(key)) {
```

When you find the function you want and check the code, you will find the part that compares the input value to a hard-coded string fragment.



Reconstructing the string with attention to offset and order will help identify the correct string.

```

λ ReverseMe.exe 5c3b8a21d79e465f0c321b987a4e65f3
CSG_FLAG{You_have_great_reversing_skills!}

```

: