

Autor:	f0rizen
Tytuł:	find a real key
Język:	C/C++
Platforma:	Unix/Linux
Trudność:	2.2
Jakość:	4.6
Arch:	x86-64
Link:	<a href="https://crackmes.one/crackme/629e1e5833c5d4261e72375f">https://crackmes.one/crackme/629e1e5833c5d4261e72375f</a>
Opis:	Find a real key and recover a flag
Plik:	<a href="https://crackmes.one/static/crackme/629e1e5833c5d4261e72375f.zip">https://crackmes.one/static/crackme/629e1e5833c5d4261e72375f.zip</a>

Krok	Rozwiązanie
1	Sprawdzenie jak działa plik .exe Działanie polega na flagi jako argumenty. Zwraca czy wartość jest poprawna czy nie.
2	Odnalezienie głównej funkcji w programie za pomocą Symbol Tree. Poznałem parę nazw żeby program był bardziej czytelny. Takich jak: parametry funkcji main iteratory inputy Efekty widać na zrzucie ekranu po prawej stronie
3	Z dłuższej analizy widać że na bazie podanej naszej flagi i wartości "sup3r_s3cr3t_k3y_1337" tworzone jest tablica z wartościami i następnie XOR tej tablicy z naszym podanym wejściem jest porównywany z tablica_intow Czyli musimy sobie ten proces odwrócić Zaczniemy od pierwszego byte'u ? - reprezentuje nieznamy dla nas znak który da dobre wyjście  s ^ 7 = 0x37 0x73 ^ 7 = 0x37  zamieniając to na binarne otrzymamy 01110011 s xxxxxxx ? 00101111 0x37  i teraz możemy odwrócić operację żeby poznać wartość ? 01110011 s XOR 01000100 ? 00101111 0x37  dodatkowo można zauważyć że jeżeli teraz użyjemy s XOR 0x37 to otrzymamy wartość ?  110111 1110011 1000100
4	Teraz możemy odwrócić proces Polecam stworzyć program w C++, żeby XOR za nas  Otrzymaliśmy flagę  File Actions Edit View Help [kali@kali:~/Desktop/629] \$ ./crackme Usage: ./crackme FLAG [kali@kali:~/Desktop/629] \$ ./crackme test Wrong flag [kali@kali:~/Desktop/629] \$ ./crackme flag{y0u_f0und_key_} You found a flag! flag{y0u_f0und_key_} [kali@kali:~/Desktop/629] \$

Zrzuty Ekranu

```

(kali@kali):~/Desktop/629
$ ./crackme
Usage: ./crackme FLAG
(kali@kali):~/Desktop/629
$ ./crackme test
Wrong flag

```

```

C:\Program Files\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\14.29.30133\bin\Hostx64-AMD64\cl.exe
1  undefined main(int argc, char * argv)
2
3
4  undefined __vfprintf
5  main_0_offset
6  long __vfprintf
7  long __vfprintf
8  int local_10
9  int local_14
10 byte sublocal_10
11 long local_18
12 long local_1c
13
14 local_1c = (long *)(&__vfprintf + 0x20);
15 if (argc == 1) {
16     puts("Usage: ./crackme FLAG");
17     return 1;
18 }
19 else {
20     argv = &argv[1];
21     if (argc - argv == 0x1) {
22         for (i = 0; i < 0x10; i = i + 1) {
23             tablica[i] = "sup3r_s3cr3t_k3y_1337"[i] - 0x22;
24         }
25         tablica_intow[0] = 0x37;
26         tablica_intow[1] = 0x37;
27         tablica_intow[2] = 0x37;
28         tablica_intow[3] = 0x37;
29         tablica_intow[4] = 0x37;
30         tablica_intow[5] = 0x37;
31         tablica_intow[6] = 0x37;
32         tablica_intow[7] = 0x37;
33         tablica_intow[8] = 0x37;
34         tablica_intow[9] = 0x37;
35         tablica_intow[10] = 0x37;
36         tablica_intow[11] = 0x37;
37         tablica_intow[12] = 0x37;
38         tablica_intow[13] = 0x37;
39         tablica_intow[14] = 0x37;
40         tablica_intow[15] = 0x37;
41         tablica_intow[16] = 0x37;
42         tablica_intow[17] = 0x37;
43         tablica_intow[18] = 0x37;
44         tablica_intow[19] = 0x37;
45         tablica_intow[20] = 0x37;
46         for (i = 0; i < 0x10; i = i + 1) {
47             if (atoi(tablica[i]) ^ argv[i]) != tablica_intow[i] {
48                 puts("Wrong flag");
49                 return 1;
50             }
51         }
52         printf("You found a flag! key: %s", argv[1]);
53         return 0;
54     }
55     else {
56         puts("Wrong flag");
57         return 1;
58     }
59 }
60
61 LAB_00010055
62 if (local_1c != (long *)(&__vfprintf + 0x20)) {
63     __vfprintf(stderr, "Assertion does not return v\n");
64     __stack_chk_fail();
65 }
66 return __vfprintf;
67

```

```

#include <iostream>
using namespace std;

char tablica[21];
char output[21];

for (int i = 0; i < 21; i = i + 1) {
    tablica[i] = "sup3r_s3cr3t_k3y_1337"[i] - 0x22;
    //cout << tablica[i];
}

for (int j = 0; j < 21; j = j + 1) {
    output[j] = tablica[j] ^ tablica_intow[j];
    cout << output[j];
}

```

```

Flag: y0u_f0und_key_
C:\Users\judez\source\repos\ConsoleApplication6\Debug\ConsoleAppli
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz
znie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```