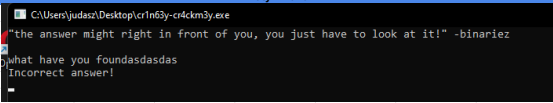
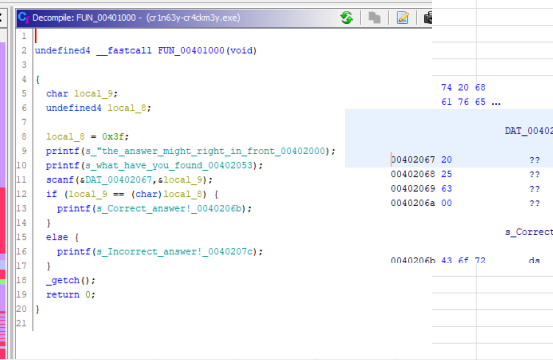
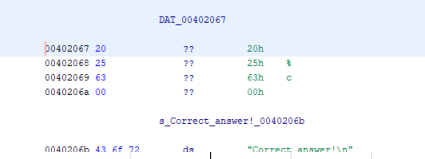
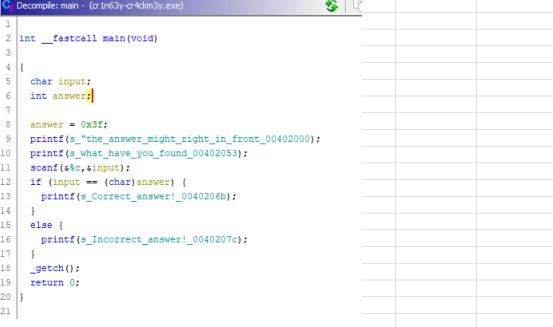
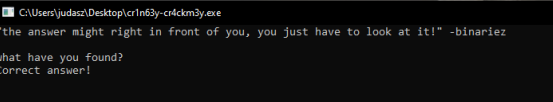


Autor:	binariez
Tytuł:	cr1n63y-cr4ckm3y
Język:	C/C++
Platforma:	Windows
Trudność:	1.4
Jakość:	3.5
Arch:	x86-64
Link:	<a href="https://crackmes.one/crackme/6309249b33c5d4425e2cd05a">https://crackmes.one/crackme/6309249b33c5d4425e2cd05a</a>
Opis:	my first crackme
Plik:	<a href="https://crackmes.one/static/crackme/6309249b33c5d4425e2cd05a.zip">https://crackmes.one/static/crackme/6309249b33c5d4425e2cd05a.zip</a>

Krok	Rozwiązanie	Zrzuty Ekranu															
1	Sprawdzenie jak działa plik .exe Działanie polega na przyjmowaniu hasła. Zwraca czy wartość jest poprawna czy nie.	 <pre>C:\Users\judasz\Desktop\cr1n63y-cr4ckm3y.exe "the answer might right in front of you, you just have to look at it!" -binariez what have you foundasdasdas Incorrect answer! "</pre>															
2	Odnalezienie głównej funkcji w programie. Funkcja <code>scanf</code> przyjmuje dwa argumenty. Sprawdzamy co się znajduje pod zmienną <code>DAT_00402067</code> . Znajduje się tam wartość <code>%c</code> . Drugi argument funkcji <code>scanf</code> to miejsce gdzie chcemy przechować nasz input. Czyli tak naprawdę sprawdzany jest pierwszy znak wprowadzony od użytkownika. Nasz <code>input</code> jest porównywany z <code>(char)local_8</code> . Wartość <code>(char)local_8</code> jest wcześniej przypisana na <code>0x3f = '?'</code> .	 <pre>Decompile: FUR_00401000 - (cr1n63y-cr4ckm3y.exe) 1 2 undefined4 __fastcall FUR_00401000(void) 3 4 { 5     char local_9; 6     undefined4 local_8; 7 8     local_8 = 0x3f; 9     printf(s_the_answer_might_right_in_front_00402000); 10    printf(s_what_have_you_found_00402053); 11    scanf(sDAT_00402067,&amp;local_9); 12    if (local_9 == (char)local_8) { 13        printf(s_Correct_answer!_0040206b); 14    } 15    else { 16        printf(s_Incorrect_answer!_0040207c); 17    } 18    _getch(); 19    return 0; 20 } 21</pre>  <table border="1"><thead><tr><th>Address</th><th>Disassembly</th><th>Comment</th></tr></thead><tbody><tr><td>00402067 20</td><td>??</td><td>20h</td></tr><tr><td>00402068 25</td><td>??</td><td>25h</td></tr><tr><td>00402069 63</td><td>??</td><td>63h c</td></tr><tr><td>0040206a 00</td><td>??</td><td>00h</td></tr></tbody></table> <p>DAT_00402067</p> <p>s_Correct_answer!_0040206b</p> <p>0040206b 43 6f 73 ds "Correct answer!\n"</p>	Address	Disassembly	Comment	00402067 20	??	20h	00402068 25	??	25h	00402069 63	??	63h c	0040206a 00	??	00h
Address	Disassembly	Comment															
00402067 20	??	20h															
00402068 25	??	25h															
00402069 63	??	63h c															
0040206a 00	??	00h															
3	Pozamieniam nazwy funkcji,zmiennych, typów zmiennych. W celu otrzymania bardziej czytelnego kodu.	 <pre>Decompile: main - (cr1n63y-cr4ckm3y.exe) 1 2 int __fastcall main(void) 3 4 { 5     char input; 6     int answer; 7 8     answer = 0x3f; 9     printf(s_the_answer_might_right_in_front_00402000); 10    printf(s_what_have_you_found_00402053); 11    scanf(s%c,&amp;input); 12    if (input == (char)answer) { 13        printf(s_Correct_answer!_0040206b); 14    } 15    else { 16        printf(s_Incorrect_answer!_0040207c); 17    } 18    _getch(); 19    return 0; 20 } 21</pre>															
4	Wartość <code>?"</code> jest prawidłowym rozwiązaniem.	 <pre>C:\Users\judasz\Desktop\cr1n63y-cr4ckm3y.exe "the answer might right in front of you, you just have to look at it!" -binariez what have you found? Correct answer!</pre>															