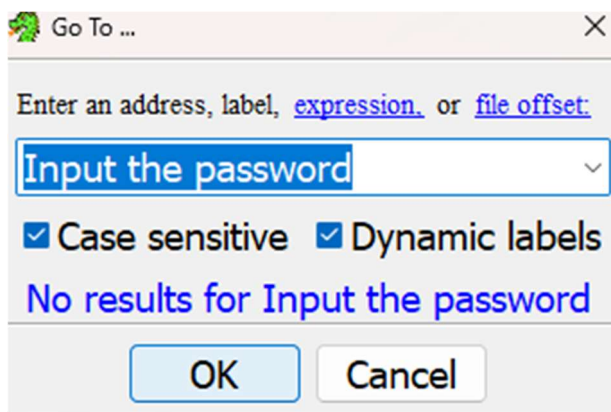
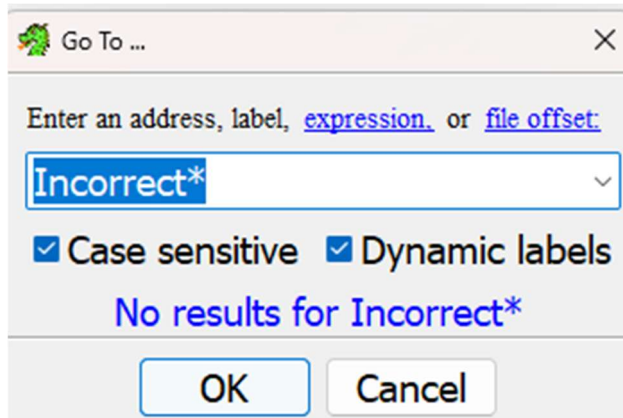


Запускаем программу, вводим пароль (конечно же) неправильно

Input the password: 123123
Incorrect

Идем смотреть в гидре. Попробуем сразу найти место проверки пароля



Но получаем увы.

Однако в процессе просмотра секций файла я заметил следующее

```
undefined4 is_it_main(void)
{
    int iVar1;
    int *piVar2;
    int *piVar3;
    int local_c8 [20];
    int aiStack_78 [17];
    char acStack_33 [20];
    char local_1f [7];
    int local_18;
    int local_14;

    FUN_00401b10();
    local_18 = 0;
    piVar2 = &DAT_00404020;
    piVar3 = local_c8;
    for (iVar1 = 0x2a; iVar1 != 0; iVar1 = iVar1 + -1) {
        *piVar3 = *piVar2;
        piVar2 = piVar2 + 1;
        piVar3 = piVar3 + 1;
    }
    for (local_14 = 0; local_14 < 0x14; local_14 = local_14 + 1) {
        putchar(local_c8[local_14] / (local_14 + 1));
    }
    scanf("%s",local_1f);
    for (local_14 = 0x14; local_14 < 0x1a; local_14 = local_14 + 1) {
        if ((int)acStack_33[local_14] == local_c8[local_14] / (local_14 + 1)) {
            local_18 = local_18 + 1;
        }
    }
    if (local_18 == 6) {
        for (local_14 = 0x1a; local_14 < 0x21; local_14 = local_14 + 1) {
            putchar(local_c8[local_14] / (local_14 + 1));
        }
    }
    else {
        for (local_14 = 0x21; local_14 < 0x2a; local_14 = local_14 + 1) {
            putchar(local_c8[local_14] / (local_14 + 1));
        }
    }
    return 0;
}
```

Проведя ряд небольших манипуляций по переименованию очевидного, получилось следующее

```

undefined4 is_it_main(void)
{
    int iVar1;
    int *piVar2;
    int *piVar3;
    int local_c8 [20];
    int aiStack_78 [17];
    char acStack_33 [20];
    char maybe_input [7];
    int local_l8;
    int i;

    FUN_00401b10();
    local_l8 = 0;
    piVar2 = &DAT_00404020;
    piVar3 = local_c8;
    for (iVar1 = 0x2a; iVar1 != 0; iVar1 = iVar1 + -1) {
        *piVar3 = *piVar2;
        piVar2 = piVar2 + 1;
        piVar3 = piVar3 + 1;
    }
    for (i = 0; i < 20; i = i + 1) {
        putchar(local_c8[i] / (i + 1));
    }
    scanf("%s", maybe_input);
    for (i = 20; i < 26; i = i + 1) {
        if ((int)acStack_33[i] == local_c8[i] / (i + 1)) {
            local_l8 = local_l8 + 1;
        }
    }
    if (local_l8 == 6) {
        for (i = 26; i < 33; i = i + 1) {
            putchar(local_c8[i] / (i + 1));
        }
    }
    else {
        for (i = 33; i < 42; i = i + 1) {
            putchar(local_c8[i] / (i + 1));
        }
    }
    return 0;
}

```

Но я пока не вижу (или не понимаю) того, что может помочь в решении задания. Пока оставим надежду найти что-то ещё в гидре, попробую найти что-нибудь через x64dbg

После запуска программы пойдём и поставим точку останова в начале модуля .exe

00401000	83EC 1C	sub esp,1C	
00401003	8B4424 20	mov eax,dword ptr ss:[esp+20]	[esp+20]:wcstombs+70
00401007	8B00	mov eax,dword ptr ds:[eax]	
00401009	8B00	mov eax,dword ptr ds:[eax]	
0040100B	3D 910000C0	cmp eax,C0000091	
00401010	77 4E	ja crackme.401060	
00401012	3D 8D0000C0	cmp eax,C0000080	
00401017	73 60	jae crackme.401079	
00401019	3D 050000C0	cmp eax,C0000005	
0040101E	0F85 CC000000	jne crackme.4010F0	

Апдейт:

Надо было внимательнее рассмотреть вызовы scanf в этом месте

004014e6	e8 bd 26	CALL	MSVCRT.DLL:scanf	26	}
00	00			27	scanf("%s",maybe_input);
004014eb	c7 84 24	MOV	dword ptr [ESP + 1],20	28	for (i = 20; i < 26; i = i + 1) {
cc	00 00			29	if ((int)acStack_33[i] == local_c8[i] / (i + 1)) {
00	14 00 ...			30	local_18 = local_18 + 1;
004014f6	eb 41	JMP	LAB_00401539	31	}
				32	}

Если поставить точку останова в данном месте в отладчике, то можно увидеть следующее:

EIP	004014EB	C78424 C0000000 14000000	mov dword ptrss:[esp+CC],14		
	004014F6	EB 41	jmp crackme.401539		
	004014F8	8B8424 C0000000	mov eax,dword ptrss:[esp+CC]		
	004014FF	83E8 14	sub eax,14		
	00401502	0FB68404 C1000000	movzx eax,byte ptrss:[esp+eax+C1]		
	0040150A	0FBEC8	movsx ecx,a1		
	0040150D	8B8424 C0000000	mov eax,dword ptrss:[esp+CC]		
	00401514	8B4484 18	mov eax,dword ptrss:[esp+eax*4+18]		
	00401518	8B9424 C0000000	mov edx,dword ptrss:[esp+CC]		
	0040151F	8D72 01	lea esi,dword ptrds:[edx+1]		
	00401522	99	cq		
	00401523	F7FE	idiv esi		
	00401525	39C1	cmp ecx,eax		
	00401527	75 08	jne crackme.401531		
	00401529	838424 C8000000 01	add dword ptrss:[esp+C8],1		
	00401531	838424 C0000000 01	add dword ptrss:[esp+CC],1		
	00401539	83BC24 C0000000 19	cmp dword ptrss:[esp+CC],19		
	00401541	7E B5	jle crackme.4014F8		
	00401543	83BC24 C8000000 06	cmp dword ptrss:[esp+C8],6		

Также увидим цикл, поставим точки останова на вызове первоначальной функции и конце цикла, смотрим регистры при вводе пароля

CPU	Журнал	Заметки	Точки останова	Карта памяти	Стек вызовов	SEH	Сценарий
	004014EB	C78424 C0000000 14000000	mov dword ptrss:[esp+CC],14				
	004014F6	EB 41	jmp crackme.401539				
	004014F8	8B8424 C0000000	mov eax,dword ptrss:[esp+CC]				
	004014FF	83E8 14	sub eax,14				
	00401502	0FB68404 C1000000	movzx eax,byte ptrss:[esp+eax+C1]				
	0040150A	0FBEC8	movsx ecx,a1				
	0040150D	8B8424 C0000000	mov eax,dword ptrss:[esp+CC]				
	00401514	8B4484 18	mov eax,dword ptrss:[esp+eax*4+18]				
	00401518	8B9424 C0000000	mov edx,dword ptrss:[esp+CC]				
	0040151F	8D72 01	lea esi,dword ptrds:[edx+1]				
	00401522	99	cq				
	00401523	F7FE	idiv esi				
	00401525	39C1	cmp ecx,eax				
	00401527	75 08	jne crackme.401531				
	00401529	838424 C8000000 01	add dword ptrss:[esp+C8],1				
	00401531	838424 C0000000 01	add dword ptrss:[esp+CC],1				
	00401539	83BC24 C0000000 19	cmp dword ptrss:[esp+CC],19				
	00401541	7E B5	jle crackme.4014F8				
	00401543	83BC24 C8000000 06	cmp dword ptrss:[esp+C8],6				

CPU	Журнал	Заметки	Точки останова	Карта памяти	Стек вызовов	SEH	Сценарий
	004014EB	C78424 C0000000 14000000	mov dword ptrss:[esp+CC],14				
	004014F6	EB 41	jmp crackme.401539				
	004014F8	8B8424 C0000000	mov eax,dword ptrss:[esp+CC]				
	004014FF	83E8 14	sub eax,14				
	00401502	0FB68404 C1000000	movzx eax,byte ptrss:[esp+eax+C1]				
	0040150A	0FBEC8	movsx ecx,a1				
	0040150D	8B8424 C0000000	mov eax,dword ptrss:[esp+CC]				
	00401514	8B4484 18	mov eax,dword ptrss:[esp+eax*4+18]				
	00401518	8B9424 C0000000	mov edx,dword ptrss:[esp+CC]				
	0040151F	8D72 01	lea esi,dword ptrds:[edx+1]				
	00401522	99	cq				
	00401523	F7FE	idiv esi				
	00401525	39C1	cmp ecx,eax				
	00401527	75 08	jne crackme.401531				
	00401529	838424 C8000000 01	add dword ptrss:[esp+C8],1				
	00401531	838424 C0000000 01	add dword ptrss:[esp+CC],1				
	00401539	83BC24 C0000000 19	cmp dword ptrss:[esp+CC],19				
	00401541	7E B5	jle crackme.4014F8				
	00401543	83BC24 C8000000 06	cmp dword ptrss:[esp+C8],6				
	00401548	75 41	jne crackme.40158E				

Address	Disassembly	Comment
004014EB	mov dword ptrss:[esp+CC],14	
004014F6	jmp crackme.401539	
004014F8	mov eax,dword ptrss:[esp+CC]	
004014FF	sub eax,14	
00401502	movzx eax,byte ptrss:[esp+eax+C1]	
0040150A	movsx ecx,a1	
0040150D	mov eax,dword ptrss:[esp+CC]	
00401514	mov eax,dword ptrss:[esp+eax*4+18]	
00401518	mov edx,dword ptrss:[esp+CC]	
0040151F	lea esi,dword ptrds:[edx+1]	
00401522	cdq	
00401523	idiv esi	
00401525	cmp ecx,eax	
00401527	jne crackme.401531	
00401529	add dword ptrss:[esp+C8],1	
00401531	add dword ptrss:[esp+CC],1	
00401539	cmp dword ptrss:[esp+CC],19	
00401541	jle crackme.4014F8	
00401543	cmp dword ptrss:[esp+C8],6	
00401548	jne crackme.401555	

Address	Disassembly	Comment
004014EB	mov dword ptrss:[esp+CC],14	
004014F6	jmp crackme.401539	
004014F8	mov eax,dword ptrss:[esp+CC]	
004014FF	sub eax,14	
00401502	movzx eax,byte ptrss:[esp+eax+C1]	
0040150A	movsx ecx,a1	
0040150D	mov eax,dword ptrss:[esp+CC]	
00401514	mov eax,dword ptrss:[esp+eax*4+18]	
00401518	mov edx,dword ptrss:[esp+CC]	
0040151F	lea esi,dword ptrds:[edx+1]	
00401522	cdq	
00401523	idiv esi	
00401525	cmp ecx,eax	
00401527	jne crackme.401531	
00401529	add dword ptrss:[esp+C8],1	
00401531	add dword ptrss:[esp+CC],1	
00401539	cmp dword ptrss:[esp+CC],19	
00401541	jle crackme.4014F8	
00401543	cmp dword ptrss:[esp+C8],6	
00401548	jne crackme.401555	

Address	Disassembly	Comment
004014EB	mov dword ptrss:[esp+CC],14	
004014F6	jmp crackme.401539	
004014F8	mov eax,dword ptrss:[esp+CC]	
004014FF	sub eax,14	
00401502	movzx eax,byte ptrss:[esp+eax+C1]	
0040150A	movsx ecx,a1	
0040150D	mov eax,dword ptrss:[esp+CC]	
00401514	mov eax,dword ptrss:[esp+eax*4+18]	
00401518	mov edx,dword ptrss:[esp+CC]	
0040151F	lea esi,dword ptrds:[edx+1]	
00401522	cdq	
00401523	idiv esi	
00401525	cmp ecx,eax	
00401527	jne crackme.401531	
00401529	add dword ptrss:[esp+C8],1	
00401531	add dword ptrss:[esp+CC],1	
00401539	cmp dword ptrss:[esp+CC],19	
00401541	jle crackme.4014F8	
00401543	cmp dword ptrss:[esp+C8],6	
00401548	jne crackme.401555	

Address	Disassembly	Comment
004014EB	mov dword ptrss:[esp+CC],14	
004014F6	jmp crackme.401539	
004014F8	mov eax,dword ptrss:[esp+CC]	
004014FF	sub eax,14	
00401502	movzx eax,byte ptrss:[esp+eax+C1]	
0040150A	movsx ecx,a1	
0040150D	mov eax,dword ptrss:[esp+CC]	
00401514	mov eax,dword ptrss:[esp+eax*4+18]	
00401518	mov edx,dword ptrss:[esp+CC]	
0040151F	lea esi,dword ptrds:[edx+1]	
00401522	cdq	
00401523	idiv esi	
00401525	cmp ecx,eax	
00401527	jne crackme.401531	
00401529	add dword ptrss:[esp+C8],1	
00401531	add dword ptrss:[esp+CC],1	
00401539	cmp dword ptrss:[esp+CC],19	
00401541	jle crackme.4014F8	
00401543	cmp dword ptrss:[esp+C8],6	
00401548	jne crackme.401555	

Введенная строка посимвольно загружается в регистр EAX, затем копируется из него в ECX. После этого в регистр EAX загружается эталонное значение проверяемого символа и по адресу 0x00401525 выполняется инструкция сравнения содержимого регистров

Получается, в регистре EAX содержатся символы пароля

Собираем их в строку, получаем слово «banana»

Проверяем

Input the password: banana
Correct

РЕЗЮМЕ:

Надо внимательнее смотреть вызовы функций чтения\записи – что записывают, куда возвращают значение, что потом с ним происходит. В начале я не придал значения инструкции scanf, что привело к бесполезному поиску инструкций валидации пароля в гидре. Надо было изучить внимательнее в дебаггере что происходит по адресу инструкции записи и куда пойдет выполнение программы далее.