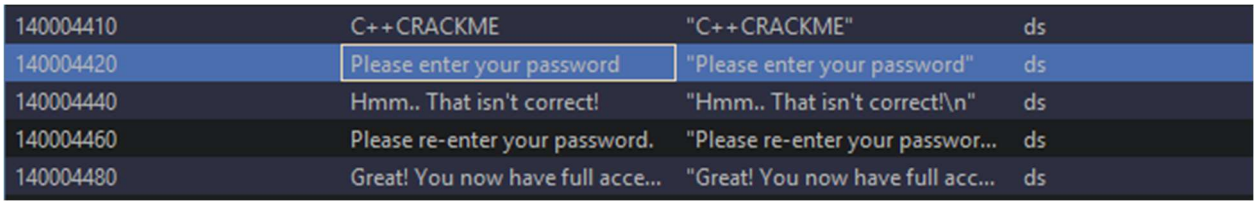
Открываем программу в гидре, ищем определенные в файле строки

Я отметил для себя как интересные все строки ниже



140004410	C++CRACKME	"C++CRACKME"	ds
140004420	Please enter your password	"Please enter your password"	ds
140004440	Hmm.. That isn't correct!	"Hmm.. That isn't correct!\n"	ds
140004460	Please re-enter your password.	"Please re-enter your passwor...	ds
140004480	Great! You now have full acce...	"Great! You now have full acc...	ds

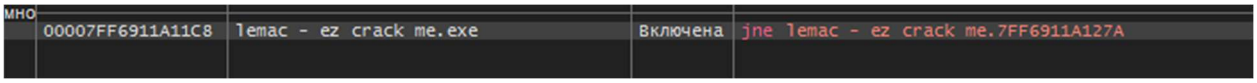
Со строкой C++CRACKME история другая

Я перешел на инструкции ввода и проверки пароля

```
this = FUN_14000lac0((basic_ostream<> *)cout_exref,"Please enter your password");
std::basic_ostream<>::operator<<(this,FUN_14000lca0);
pbVar5 = FUN_14000ldb0((basic_istream<> *)cin_exref,(longlong *)input);
bVar1 = pbVar5[(longlong)*(int *)(*(longlong *)pbVar5 + 4) + 0x10];
uVar2 = local_40;
uVar3 = local_20;
while (local_40 = uVar2, local_20 = uVar3, ((byte)bVar1 & 6) == 0) {
  _Buf2 = standart;
  if (0xf < local_18) {
    _Buf2 = (undefined8 ***)standart[0];
  }
  _Buf1 = input;
  if (0xf < local_38) {
    _Buf1 = (undefined8 ***)input[0];
  }
  _Size = uVar3;
  if (uVar2 < uVar3) {
    _Size = uVar2;
  }
  uVar4 = 0;
  if (_Size != 0) {
    uVar4 = memcmp(_Buf1,_Buf2,_Size);
  }
  if (uVar4 == 0) {
    if (uVar2 < uVar3) {
      uVar4 = 0xffffffff;
    }
    else {
      uVar4 = (uint)(uVar2 != uVar3);
    }
  }
  if (uVar4 == 0) break;
  FUN_14000lac0((basic_ostream<> *)cout_exref,"Hmm.. That isn\'t correct!\n");
  FUN_14000lac0((basic_ostream<> *)cout_exref,"Please re-enter your password.\n");
```

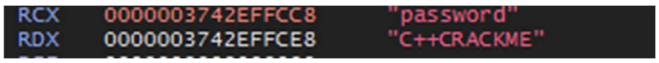
Судя по всему, пароль уже лежит в памяти и введенный пароль с ним сравнивается. Так как используется memcmp, то я посчитал что все сравнивается на уровне регистров и отдельно в программный код никуда не выводится. Поэтому я пошел смотреть, что происходит в регистрах во время цикла while

Я поставил точку останова на адрес начала цикла сравнения



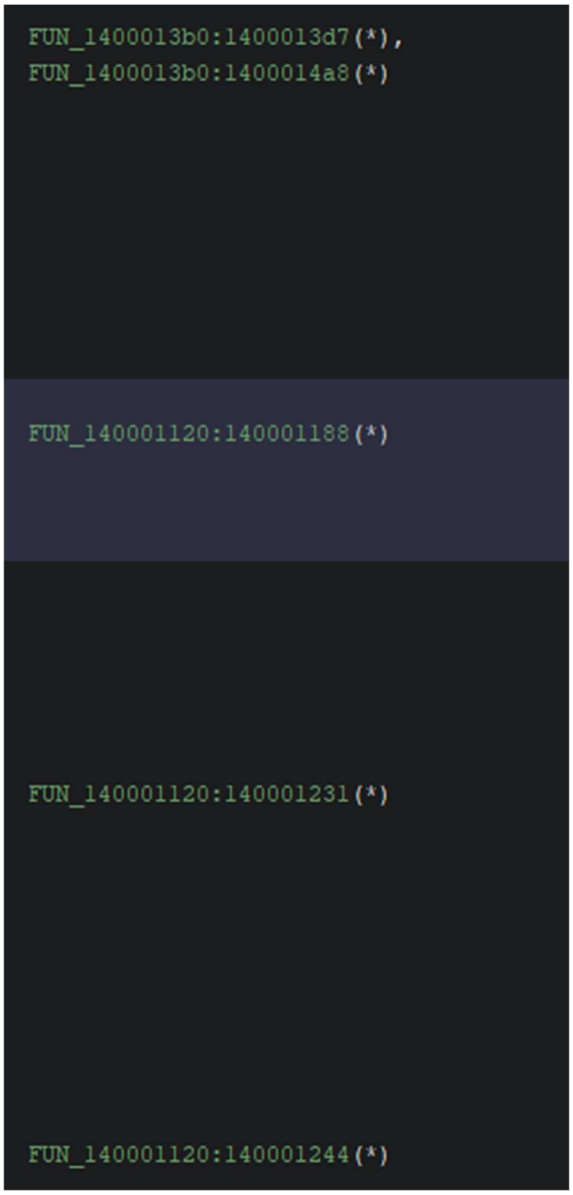
И запустил программу

И, соответственно, в регистрах появилось



Вот и пароль

А теперь к самому интересному. На самом деле строка с паролем была определена ещё в гидре, однако в отличие от других строк она не используется в одной с ними функции



Посмотрим, что происходит в xref-функции, работающей со строкой пароля.

Честно говоря, я так и не особо понял, что там происходит, зато увидел, что эта функция вызывается в main

```
FUN_1400012f0(standart,'\0',0);
FUN_1400013b0((longlong *)standart,uVar6,10);
local_40 = 0;
local_38 = 0;
FUN_1400012f0(input,'\0',0);
```

Но сильно раньше, чем вообще производится ввод пароля пользователем. Видимо, на этом этапе, судя по переданным в функцию параметру – куда, откуда, сколько, - производится композиция эталонного пароля из памяти в область локальных переменных. Да, кажется, до меня все-таки дошло, что там происходит.

Ну и, конечно, проверяем найденный пароль