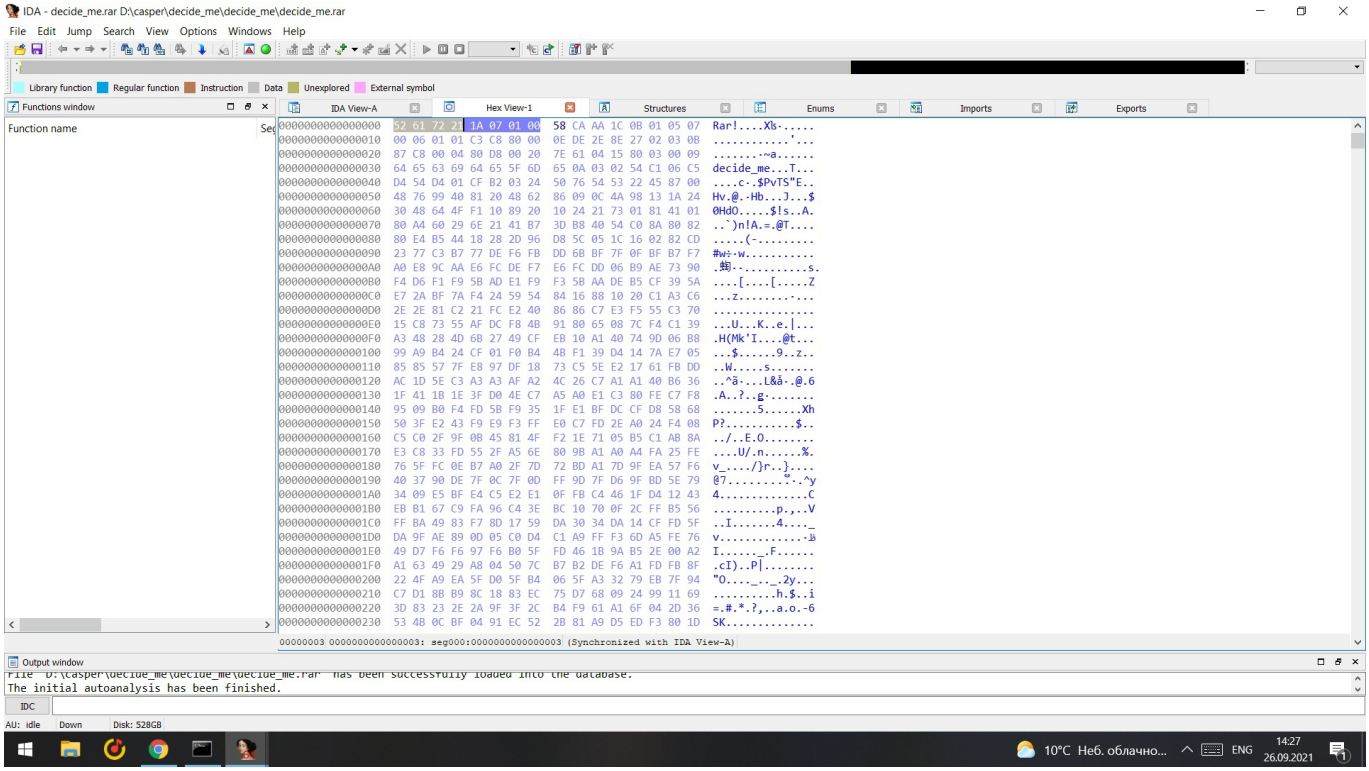
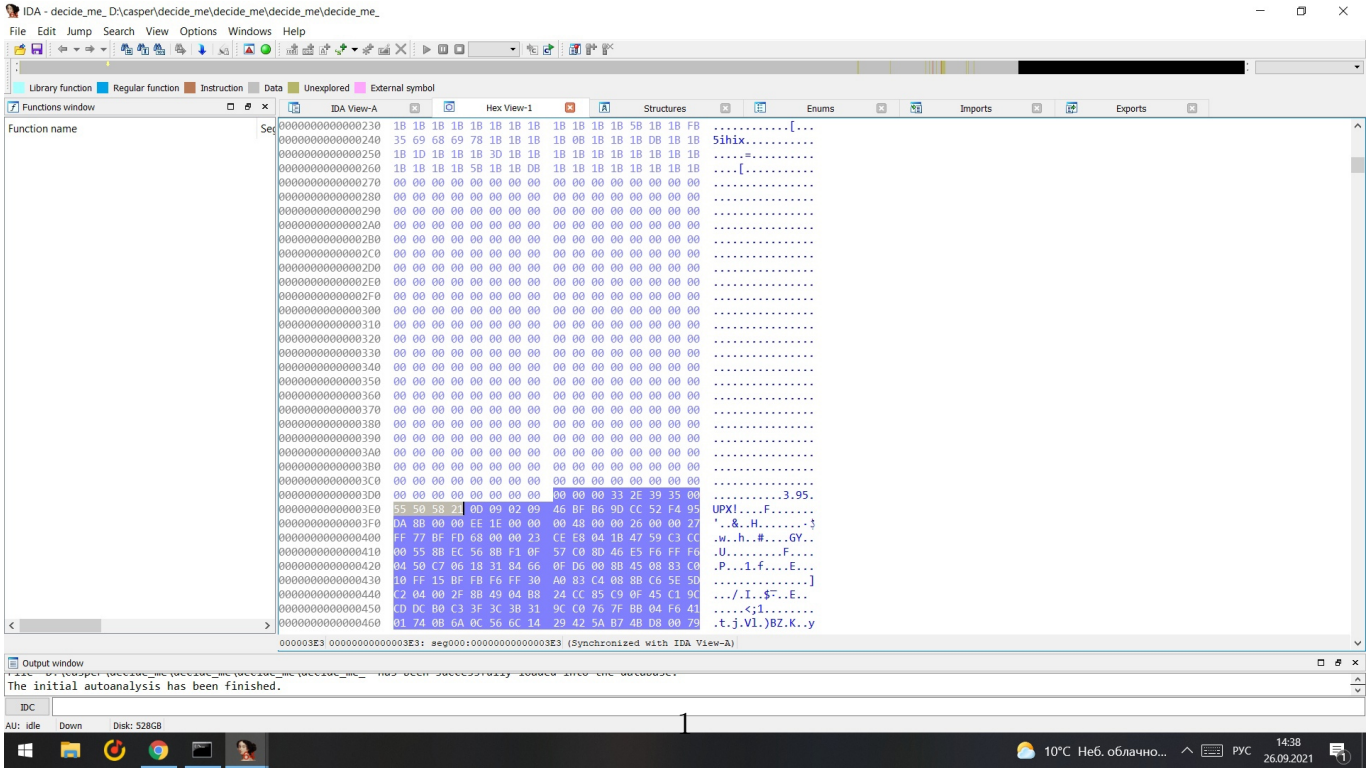
Задача на стажировку decide me

Макаров Глеб Евгеньевич

В скачанном архиве находится exe файл. Но он не запускается. Открываем его в дизассемблере Ida, видим, что первые байты бинарного кода соответствуют сигнатуре rar архива.



Меняем разрешение файла на .rar. В полученном архиве содержится один файл *decide_me* без разрешения. Открываем его в Ida, просматриваем байт код и видим, что в начале идет что-то странное, потом идут нули, и потом строчка *UPX*!.



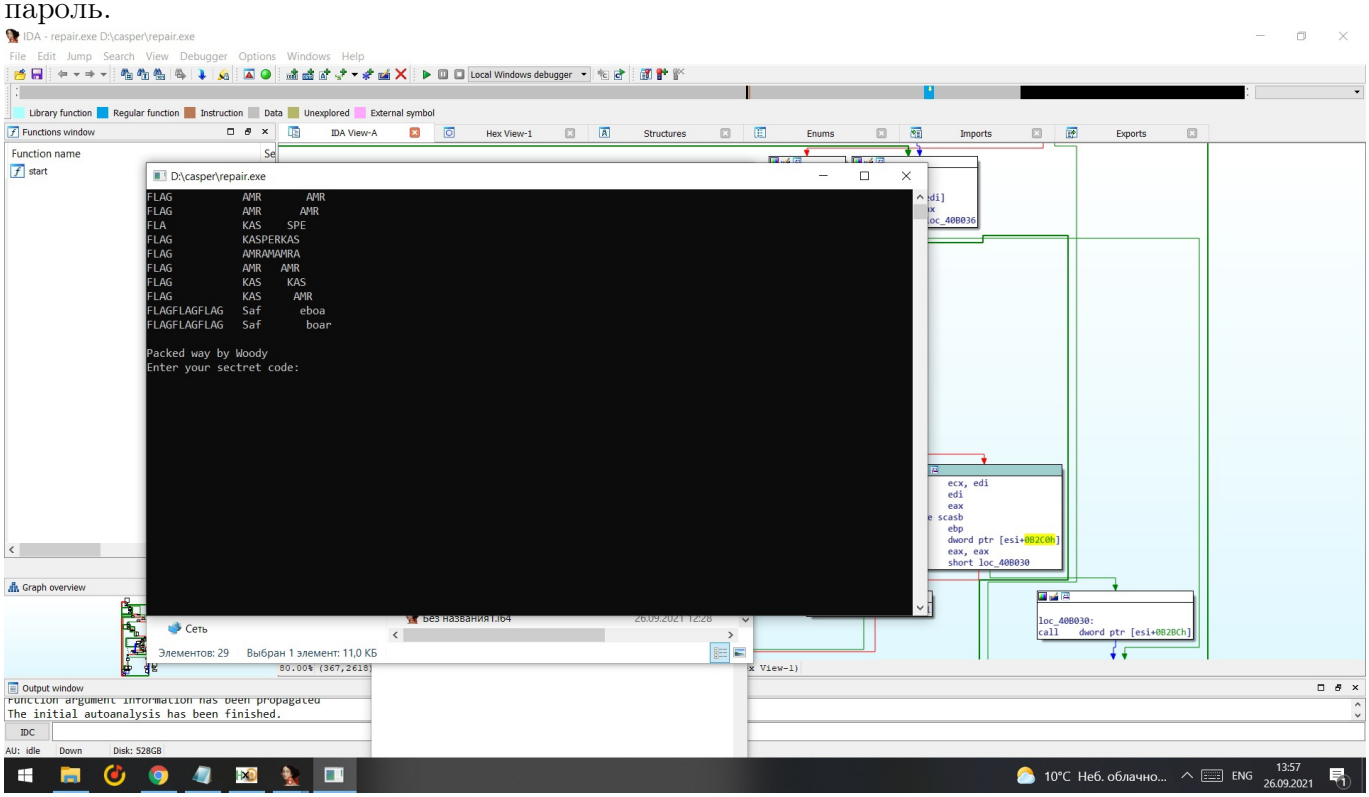
Понимаем, что перед нами файл, сжатый через UPX. Дальше я сжал простую программу через UPX, чтобы сравнить зашифрованный файл с обычным. Увидел, что в незашифрованном файле в начале есть надпись

$$This\ program\ cannot\ be\ run\ in\ DOS\ mode$$

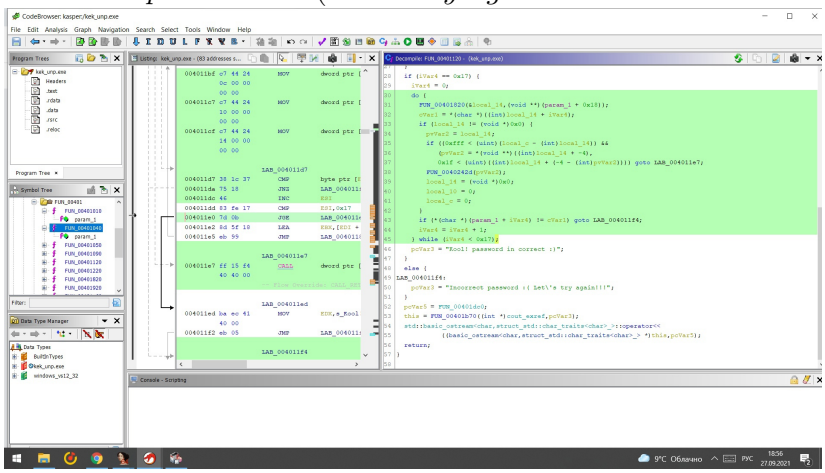
а в зашифрованном

$$Osrh; kit|izv; xzuuto; y\ ; inu; ru; \_TH; vt. \sim$$

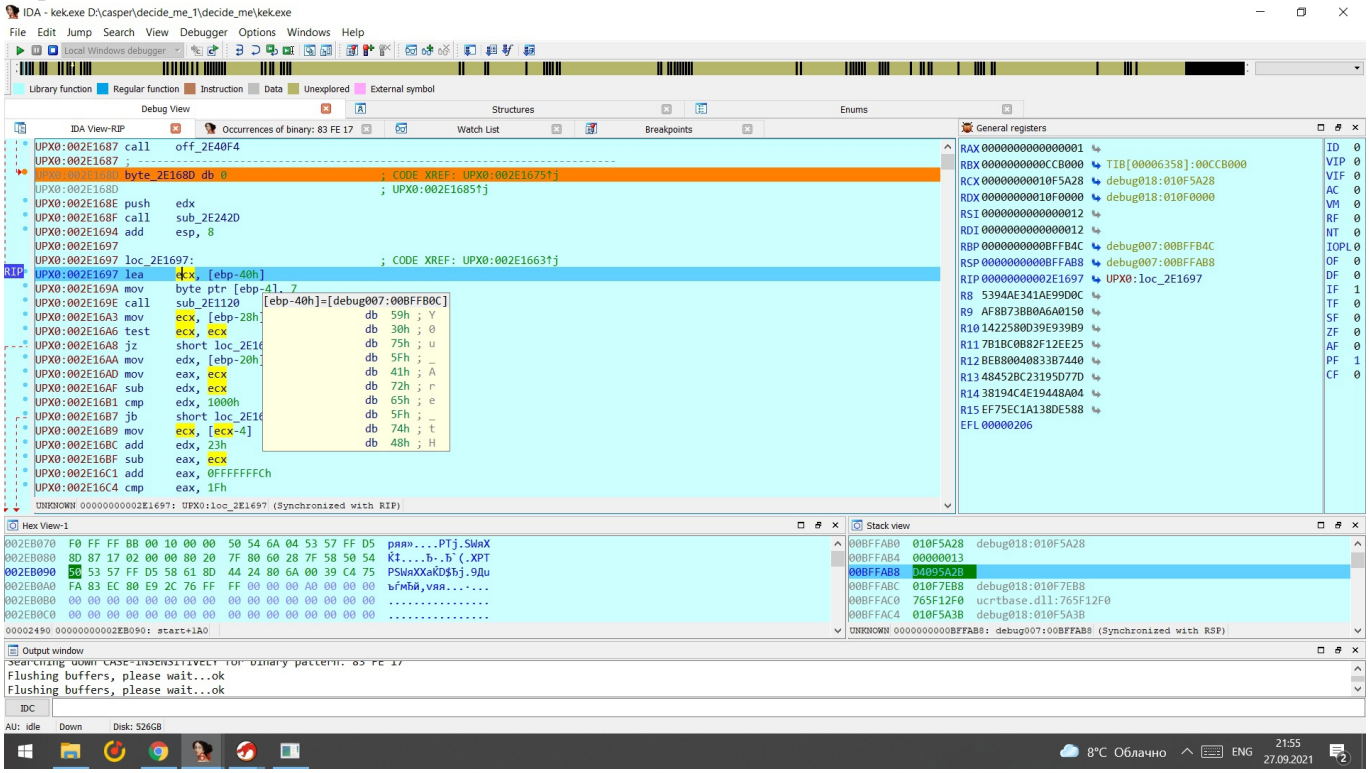
Сопоставил буквы из надписей. Путем долгого пристального взгляда понял, что байты сксорили с 0x1B(т.к. ноль превратился в 0x1B). Написал небольшую программу, чтобы снова сксорить зашифрованные байты с 0x1B. В Hex редакторе создал новый бинарник, с расшифрованными байтами. Запустил этот файл и он оказался рабочим! Программа просила ввести пароль.



Открыл файл в Ghidra и в Ida, но понял, что мало что понятно, т.к. файл сжат в UPX и ни Ida, ни Ghidra, не могут разделить программу на функции, поэтому разархивировал файл и снова открыл в Ghidra. Теперь функций, которые распознала Ghidra стало больше. Прошелся по функциям и нашел одну, которая использовала строки $Kool!$ $password\ in\ correct$:) и $Incorrect\ password$:($Let's\ try\ again$!!!.



Понял, что именно здесь программа работает с введенной мной строкой и скорее всего сравнивает ее с настоящим паролем. Ставим брейкпоинт в этом месте и запускаем дебагер, находим место, где сравниваются строки и находим саму строку, с чем сравнивается введенный пароль.



Ответ:

$$Y0u\_Are\_tHe\_bEsT\_Intern$$