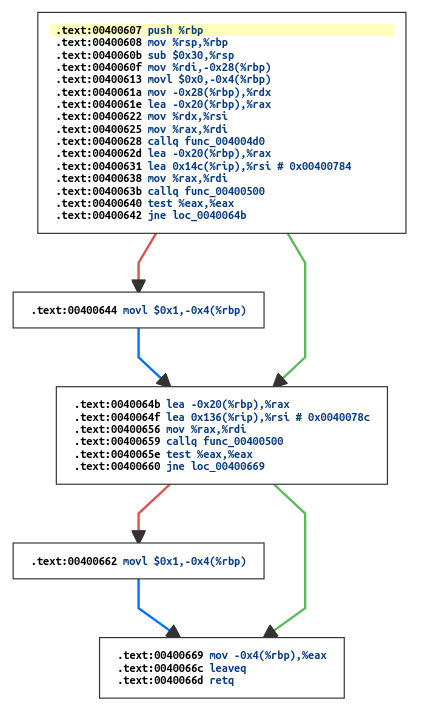
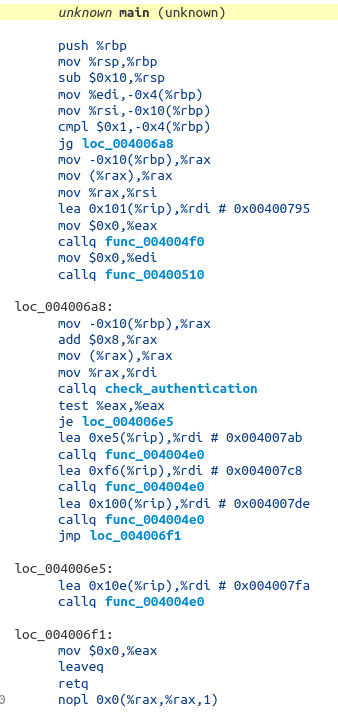
auth\_overflow — 1

Декомпилируем файл, находим функции отвечающие за работу программы, main и chek\_authentication

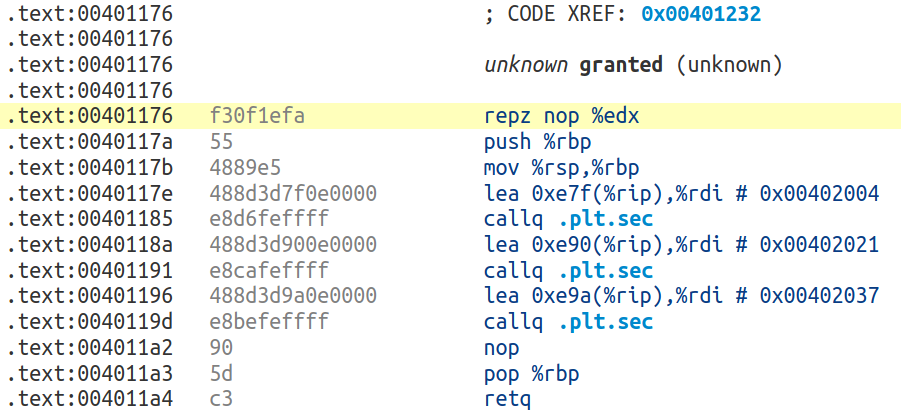


Далее по коду отслеживаем вызовы нужных нам операторов и выясняем, что из функции chek\_authentication возвращается значение ячейки -0x4 в которую мы попадаем при успешном сравнении в

test %eax, %eax

то есть нам надо добиться, чтобы байты были отличны от 0, чтобы код пустил нас в секцию с флагом, так же из работы с памятью в функции аутентификации видно, что работа идет со значениями 0х20, 0х4, 0х28, 0х14, путем нехитрой математики выясняем, что для переполнения буфера достаточно набрать 29+ символов в первом аргументе при запуске программы, чтобы она не смогла корректно отработать буфер, т. к. при компиляции с тэгом -fno-stack-protector, нет соответствующей защиты и логика работы программы нарушается.

Далее вторая задача, имеет усложненный вид, в ней пароль вводится соответственно через клавиатуру, а так же необходимо попасть в отдельную функцию granted



Для выполнения задания была использована статья

<https://habr.com/ru/post/266591/>, адресс функции granted 0x4005A7 и он же в виде символов «.@.....» с помощью программы IDA Pro. Далее необходимо подставить 24 символа и адрес функции granted чтобы с помощью используемой ранее уязвимости переполнения буфера попасть в функцию granted, без используемых выше флагов компиляции данное действие невозможно.

Далее скриншоты функций main и chek\_authentication, пришлось немного сжать, чтобы влезли на одну страницу,

