Собрал пример из урока (под MAC OS).

Вначале посмотрел проверку пароля в функции main:

100000eca: e8 71 ff ff ff callq -143 <\_checkPass>

100000ecf: a8 01 testb $1, %al

100000ed1: 0f 85 05 00 00 00 jne 5 <\_main+0x5c>

100000ed7: e9 16 00 00 00 jmp 22 <\_main+0x72>

100000edc: 48 8d 3d a8 00 00 00 leaq 168(%rip), %rdi

Как видим при верном пароле мы переходим на строку 0edc, а при неверном прыгаем на 22 байта дальше. Первым делом я заменил пять байт jmp на пять команд nop:

100000ecf: a8 01 testb $1, %al

100000ed1: 0f 85 05 00 00 00 jne 5 <\_main+0x5c>

100000ed7: 90 nop

100000ed8: 90 nop

100000ed9: 90 nop

100000eda: 90 nop

100000edb: 90 nop

100000edc: 48 8d 3d a8 00 00 00 leaq 168(%rip), %rdi

Е-но программа стала принимать любой пароль, но мне не понятно зачем вообще мы запаролили программу, поэтому я пошел в начало функции и нашел парвый call:

100000e80: 55 pushq %rbp

100000e81: 48 89 e5 movq %rsp, %rbp

100000e84: 48 81 ec 90 00 00 00 subq $144, %rsp

100000e8b: 48 8d 3d e0 00 00 00 leaq 224(%rip), %rdi

100000e92: 48 8b 05 77 01 00 00 movq 375(%rip), %rax

100000e99: 48 8b 00 movq (%rax), %rax

100000e9c: 48 89 45 f8 movq %rax, -8(%rbp)

100000ea0: c7 45 8c 00 00 00 00 movl $0, -116(%rbp)

100000ea7: b0 00 movb $0, %al

100000ea9: e8 88 00 00 00 callq 136

100000eae: 48 8d 3d d3 00 00 00 leaq 211(%rip), %rdi

Думаю будет логично не запрашивать пароль а перейти сразу к делу!

Команда jmp как и команда call занимает 5 байт, значит чтобы попасть сразу на ячейку edc надо отнять от её адреса eae итого: надо заметить call на jmp 2E

100000ea9: e9 2e 00 00 00 jmp 46 <\_main+0x5c>

100000ed1: 0f 85 05 00 00 00 jne 5 <\_main+0x5c>

Адрес <\_main+0x5c> совпадает с адресом в jne значит я все правильно посчитал смещение, теперь программа не задает лишних вопросов и сразу переходит к делу!