1 уровень. Ввести строку длины больше 127 символов. Максимально возможное значение переменной типа char 127, а функция strlen возвращает unsigned int. И когда мы вводим строку длины больше 127, происходит переполнение и значение переменной становится отрицательным.

2 уровень. 4278255871 65535 255 Выражение (ui+us\*uc) приводится к максимальному используемому в нем типу(unsigned int). И чтобы получился 0, нужно чтобы все 32 бита были нулями. Первое число, выходящее за пределы unsigned int и имеющее 32 бита равные нулю, это 2^32.

3 уровень. -201 Чтобы получилось число больше 200, нужно ввести отрицательное число. Функция atoi() возвращает знаковый int, который может быть отрицательным. При преобразовании отрицательного числа мы получим переполнение и, как результат, i > 200.

4 уровень. 4294967294 Чтобы полученное число стало равно нулю, аналогично второму уровню нужно чтобы все 32 бита итогового числа были нулями. То есть это 2^32. Значит ввести нужно число 2^32-2, так как в процессе выполнения программы к введенному числу прибавляется 2.

5 уровень. Нужно ввести строку из >=16 символов Переменные buf и a статические, поэтому под них память выделяется на стеке. Выход за пределы массива не контролируется в языке Си, мы можем ввести строку длины, большей чем 16, произойдет выход за пределы массива, и тем самым перепишем значение переменной a, так как она находится в стеке перед buf.

6 уровень. System32 Ввести название папки, в которой находится файл cmd.exe

7 уровень. 0000000000000000 Ввести строку из 16 символов. Размер буфера у нас 16 байт, но последний занимает нулевой символ, так как в языке Си строки нуль-терминированные. А мы ввели 16 символов. Значит они займут весь буфер и нулевой символ запишется в стедующий байт в стеке, который является последним байтом переменной b.

8 уровень. "%x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x" Ввести строку, задающую формат вывода. Так как в программе вызывается функция printf(argv[1]); без аргументов для форматирования, то на выводе окажутся значения, лежащие на стеке. Таким образом, мы сможем получить нужное нам значение (выведется оно перед 0xaaaaaaaa, так как было добавлено туда сразу после него).

9 уровень. 22222222222 22222222222222222222222222222222 Пароль должен состоять из 32 символов, чтобы при копировании аргумента командной строки в символьный массив password последний символ оказался не терминирующим нулем. В результате strcat(key,password); не встретит терминирующий ноль и будет неконтролируемо перезаписывать данные в стек, где находится в том числе и переменная i.

10 уровень. D:\СП\levels\level10.exe Передаем полный путь к файлу и больше ничего. Его длина больше 16 символов. Сначала в цикле в buf записываются первые 16 символов, после записи 16-го символа i=16. И дальше buf[16](которое является 17-м символом, т.е. произошел выход за пределы массива buf) присваивается 0, а по этому адресу как раз располагается первый байт переменной i, равный в данный момент 16. Ему и присваивается значение 0.

11 уровень. qwertyuqwertyhhgfdsasdfghgfdsasdfghjkjhgfdssdfghjkkjhgfdsdfghjkjhgfddfghjjhgfddffghfDCBA Нужно ввести строку из 88 символов, последние 4 из которых равны DCBA. Оно занимает 4 байта. В стеке располагаются в указанном порядке: содержимое массива buf, которое занимает 80 байт, переменная saved\_ebp, которая занимает 4 байта, и переменная saved\_eip, которая также занимает 4 байта и в которой нам нужно получить значение 0x41424344. Значит если мы введем строку из 88 символов, то первые 80 символов заполнят массив buf, следующие 4 символа заполнят переменную saved\_ebp, и оставшиеся 4 символа будут записаны в переменную saved\_eip. Так как каждый из 4 байт значения переменной saved\_eip перезаписывается кодом введенного нами символа, то, обратившись к таблице кодов ASCII, находим, что код 0x41 соответствует символу A, код 0x42 - символу B, код 0x43 - символу C и код 0x44 - символу D. И так как в стеке байты располагаются от младшего байта к старшему, то вводить символы нужно в обратном порядке, т.е. DCBA.

12 уровень. Строка: 84 любых символа и последние 4 символа следующие: ctrl+O, ctrl+P, @, null(ctrl+2), т.е. всего 88 символов. Если мы запустим программу, можно заметить вывод адреса точки входа в функцию unreachable(). Переведем этот адрес(int) в строку символов (4\*char) (Получим символы с кодами 0, 64, 32, 15). По таблице ASCII эти символы равны null, @, ^P, ^O соответственно. Эти байты в обратном порядке нужно записать в адрес возврата main, чтобы после выхода из неё запустилась функция unreachable. Но в стеке находятся в следующем порядке: buf размером 80 байт, аргументы main размером 4 байта и адрес возврата размером 4 байта. То есть чтобе перезаписать адрес возврата, нужно также перезаписать еще и аргументы main, то есть понадобится 80+4+4=88 символов.