Лабораторная № 4, Шинкевич Дарья, 15 группа

Level 1

Длина строки присваивается в переменную типа char, размером 1 байт. Ее диапазон значений от -128 до 127. Передадим строку длины 128, произойдет переполнение, в переменной с окажется отрицательное число.

Level 2

Передаем 4 294 967 294, 1 и 258

Тогда uc = 2, так как произошло переполнение и в переменную записан остаток от деления на 256.

При вычислении ui + us \* uc результат будет иметь тип большего операнда – unsigned int. Но при вычислении получается 4294967294 + 1 \* 2 = 4294967296, а в unsigned int могут быть значения от 0 до 4294967295. 4294967296 в двоичной системе

1 00000000 00000000 00000000 00000000, но учитываются только младшие 32 бит, поэтому результат 0.

Level 3

Передать -1. Тогда при присвоении в unsigned int переменную -1 записывается как

-1+2^32 = 4294967295, которое больше 200.

Level 4

Передаем 4294967294. Это значение больше INT\_MAX, поэтому strtol вернет INT\_MAX = 2147483647 и a будет неотрицательным. При этом i + 2 должно быть равно 4294967296, но максимальное значение для unsigned int – 42949672995. Поэтому в s запишется 0.

Level 5

Функция gets является небезопасной, поскольку никак не проверяется количество считываемых символов. Если размер считываемых данных больше буфера для записи, запись продолжается, но уже в память, для этого не предназначенную. В программе выделяется память на массив из 16 символов, ввести 16(тогда “не помещается” ‘\0’) или больше символов, и следующая переменная в стеке – а – будет перезаписана.

Level 6

В качестве аргумента передать System32, так как cmd.exe находится по адресу C:\Windows\System32\cmd.exe.

Level 7

В качестве аргумента передать строку длиной 16 символов. В действительности она будет содержать 17 символов, т.к. в конце строки добавится ‘\0’. Нуль-терминатор не помещается в строку, и в результате записывается в младший байт следующей переменной – переменной b. Изначально в b находилось значение 2^32 (111…1 – 32 единицы в двоичной системе). После записи массива символов, ‘\0’ с кодом ноль запишется в младший байт b.

Тогда b = 111111111111111111111111000000002= ffffff0016

Level 8

Введем “%x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x %x”.

Printf ищет аргументы в шестнадцатеричной форме в стеке, но так как никакие аргументы не были переданы, то printf воспринимает как аргументы находящиеся в стеке адрес возврата, адрес строки формата и т.д. 20 аргументом будет сгенерированное случайное значение i. Его и введем как искомое случайное.

Level 9

Вводим abcdef abcdefghijabcdefghijabcdefghijab (в пароле 32 символа). Тогда в key записывается “key=” и 32 символа пароля и символы окончания строки. Получившаяся строка не помещается в память, выделенную под key и символы пароля записываются в переменную i, так как strcat никак не проверяет размер итоговой строки.

Level 10

Запускаем файл без передачи аргументов, используя полный путь к exe файлу. Нулевым аргументом командной строки как раз и будет полный путь к файлу level10.exe, и его длина больше 16. Тогда при выходе из цикла i = 16 при попытке записи в элемент массива с индексом 16 перезаписывается локальная переменная i.

Level 11

Под buf выделяется 80 байт памяти, под saved\_ebp и saved\_eip по 4. Соответственно, чтобы перезаписать saved\_eip надо в качестве аргумента передать строку длиной 88 стмволов. Первые 80 символов произвольные, они запишутся в buf. Следующие 4 тоже произвольные, запишутся в saved\_ebp, т.к. она создавалась перед buf. Коды последних 4 символов запишутся в saved\_eip

4142434416 = 01000001 01000010 01000011 010001002

010000012 = 6510 – символ A

010000102 = 6610 – символ B

010000112 = 6710 – символ C

010001002 = 6810 – символ D

Последними символами должны быть DCBA(в обратном порядке, т.к. страшие байты переменной находятся глубже в стеке)

Level 12

Printf выводит 0x40100f – это адрес функции unreachable. В двоичной системе это –

00000000 01000000 00010000 00001111

000000002 – 010

010000002 – 642

000100002 – 162

000011112 – 152

В соответствии с таблицей ASCII для этих символов нужно ввести Ctrl+@, @, Ctrl+P, Ctrl+O. Но в обратном порядке, так как в стеке сначала идут младшие байты числа.

В стеке находятся 80 символов из buf(80 байт), аргументы main(4 байт) и адрес возврата. Поэтому введем 84 произвольных символа, а дальше Ctrl+O, Ctrl+P, @, Ctrl+@.