

При запуске программы видим запрос пароля, и, естественно, неверный статус введенного случайного набора символов

```
Please~e, can~n~n you writew pas~s~sword <3 :
asdas
N~n~nop, you~u~u fail~l~led :(((
```

По вхождению строки находим функцию, в которой она выводится:

```
__main();
password = (char *)malloc(0x400);
puts("Please~e, can~n~n you writew pas~s~sword <3 :");
scanf("%s",password);
char1 = 0x36617ab5;
char2 = 0x36617a9e;
char3 = 0x36617ab0;
char4 = 0x36617a95;
char5 = 0x36617ab6;
char6 = 0x36617a9b;
char7 = 0x36617abb;
char8 = 0x36617a84;
char9 = 0x36617ac1;
char10 = 0x36617ace;
char11 = 0x36617ac1;
char12 = 0x36617ace;
char13 = 0x36617ac6;
char14 = 0x36617ac4;
char15 = 0x36617ac4;
char16 = 0x36617ac5;
char17 = 0x36617ac5;
char18 = 0x36617ac5;
char19 = 0x36617acf;
is_valid_pwd = cmp_pass(password, (longlong)&char1);
int_pwd_is_valid = (int)is_valid_pwd;
if (int_pwd_is_valid == 0) {
    puts("N~n~nop, you~u~u fail~l~led :(((");
}
else {
    puts("Ye~e~eah, you~u, righ~h~ht ;)))");
    puts("there~e codesw from nuclearw bombs~s: ");
    printf("%d\t%d\n%d\t%d", 0x26alf, 0xe9f8f, 0x793lc, 0x62f65);
}
```

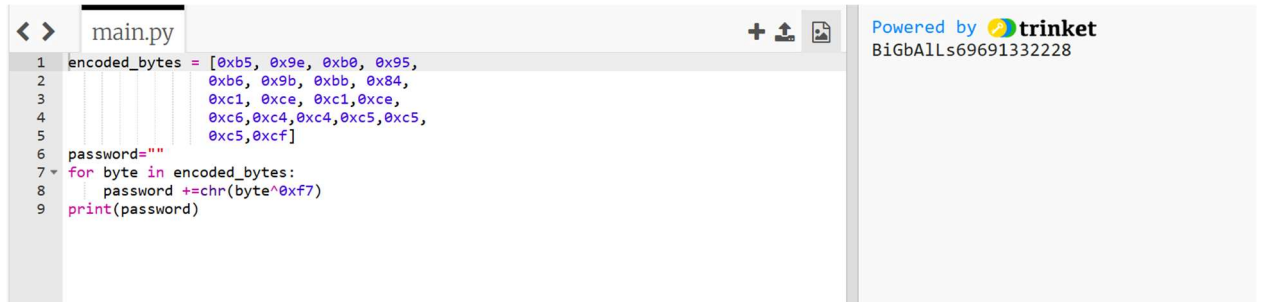
Видим набор переменных, и функцию cmp_pass.

Переходим в неё


```
PwdLength = strlen(param_1);
if (PwdLength == 19) {
    for (i = 0; i < 19; i = i + 1) {
        if (param_1[i] != (byte)((byte)*(undefined4 *) (param_2 + (ulonglong)i * 4) ^ 0xf7)) {
            return 0;
        }
    }
}
```

Значит, что происходит? Есть массив переменных, в функцию они передаются по указателю на начало массива, причем их значение конвертируется в байтовое, то есть используется не все записанное значение, а два последних – в x86-64 порядок записи с младшим значащим битом, значит мы должны взять младший байт значения и xor 0xf7

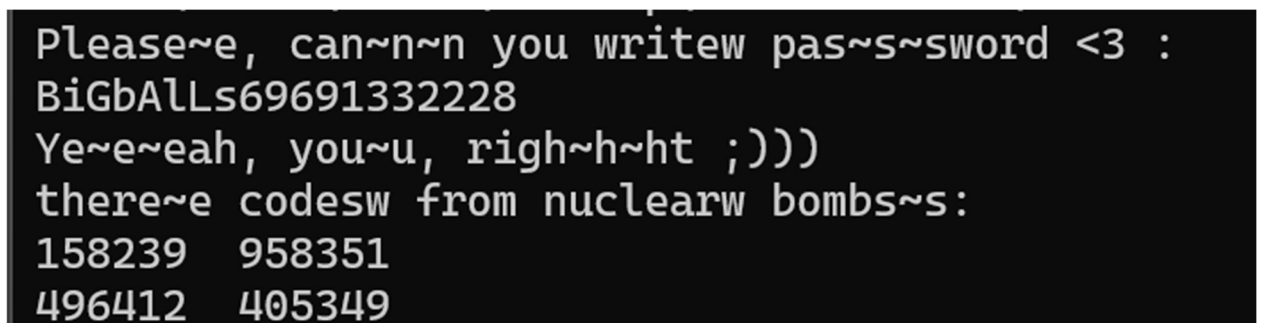
Пишем небольшой скрипт на питоне, получаем пароль



```
1 encoded_bytes = [0xb5, 0x9e, 0xb0, 0x95,  
2                 0xb6, 0x9b, 0xbb, 0x84,  
3                 0xc1, 0xce, 0xc1, 0xce,  
4                 0xc6, 0xc4, 0xc4, 0xc5, 0xc5,  
5                 0xc5, 0xcf]  
6 password=""  
7 for byte in encoded_bytes:  
8     password += chr(byte^0xf7)  
9 print(password)
```

Powered by  trinket
BiGbAlLs69691332228

Проверяем



```
Please~e, can~n~n you writew pas~s~sword <3 :  
BiGbAlLs69691332228  
Ye~e~eah, you~u, righ~h~ht ;)))  
there~e codesw from nuclearw bombs~s:  
158239 958351  
496412 405349
```

Всё верно