Лабортаторна робота №2

Підгрупа 2В. Бібліотека PyCrypto під Linux платформу

Варто почати з того, що й попередній лабораторній, роботі, наразі PyCrypto застрів, з ним виникають проблеми, тож доцільніше використовувати Pycryptodome. Отож, розглянемо основні методи генерації ПВЧ в Python.

random.random():

Ця функція не є криптографічно стійкою, вона написана на мові с.

```
static PyObject *
_random_Random_random_impl(RandomObject *self)
/*[clinic end generated code: output=117ff99ee53d755c input=afb2a59cbbb00349]*/
{
    uint32_t a=genrand_int32(self)>>5, b=genrand_int32(self)>>6;
    return PyFloat_FromDouble((a*67108864.0+b)*(1.0/9007199254740992.0));
}
```

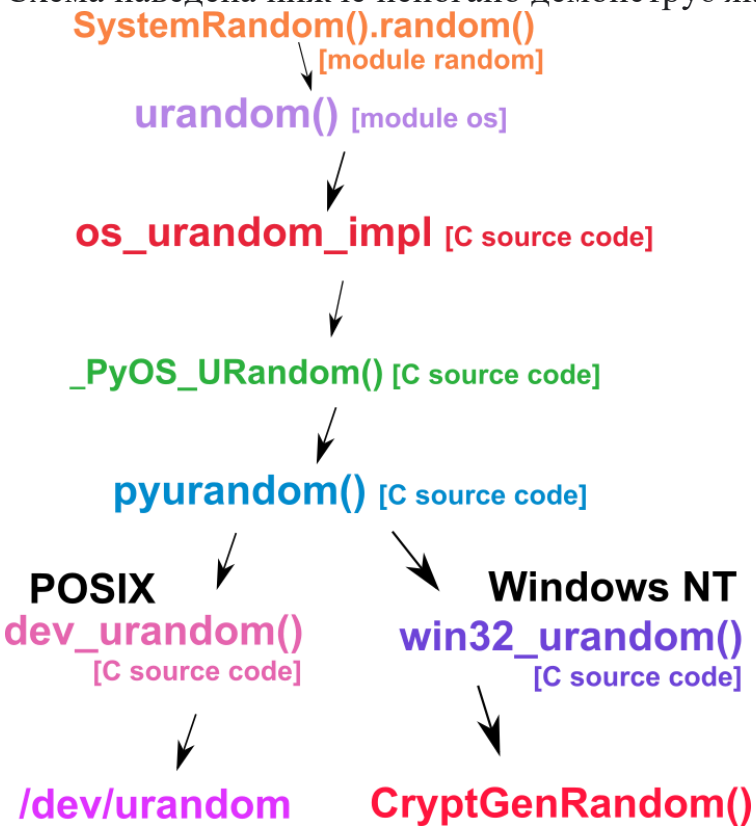
genrand_int32() — це функція, визначена генератором Mersenne Twister, яка повертає 4-байтове число. Цей алгоритм відомий своїм достатньо довгим періодом, тому він генерує якісні ПВЧ для звичайних додатків, але не для задач криптографії.

До речі, назва випливає з того факту, що в якості довжини періоду викоритовуються прості числа Марсена.

SystemRandom()

Цей генератор ПВЧ є криптографічно стійким, для ос Linux використовує /dev/urandom, у системі Windows – функцію CryptGenRandom(). В обох випадках код написаний на С.

Схема наведена нижче непогано демонструє як саме влаштована ця функція:



Тут в обох випадках, вихідні коди трошки більш заплутані тож детальна розповідь не є доцільною. Так чи інакше у контексті python використовується urandom, тож її розглянемо детальніше.

```
import secrets
import time
import os
import random


def random_method(x):

    start_time = time.time()
    random_number = []
    for _ in range(x):
        random_number.append(random.randrange(1, 20, 1))\
```

```
    end_time = time.time()
    print(end_time - start_time)


def urandom_method(y):
    start_time = time.time()
    random_number = []
    for _ in range(y):
        random_number.append(os.urandom(2))
    end_time = time.time()
    print(end_time - start_time)


random_method(10000)
urandom_method(10000)
```

```
C:\Users\Ivan_Wizard\PycharmProjects\KPI_Labs\venv\Scripts\python.exe C:/Users/Ivan_Wizard/PycharmProjects/KPI_Labs/random_tets.py
0.005984783172607422
0.0019960403442382812
```

Отже, можна сказати, що генерація ПВП у бібліотеці Pycryptodome нарізі є безпечною.