Федеральное агентство связи

Ордена трудового Красного Знамени федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Отчет по лабораторной работе

«РЕАЛИЗАЦИЯ ПРОСТЕЙШЕГО ГЕНЕРАТОРА ПАРОЛЕЙ»

по дисциплине «Информационная безопасность»

Вариант 1

Выполнил: студент группы БВТ1904

Абакаров Гасан Гаджирабаданович

Проверила:

Магомедова Дженнет Исламутдиновна

Москва, 2020

**Цель работы:** получение основных теоретических сведений и практических навыков по оценке стойкости парольной защиты.

**Ход работы:**

1. Ознакомиться с теоретической частью данной работы.

2. Составить программу-генератор паролей.

3. Составить отчет по проделанной работе.

4. Защитить работу.

**Постановка задачи**:

Требуется реализовать простейший генератор паролей, обладающий основными требованиями к парольным генераторам. Программа должна выполнять следующие действия.

1. Ввод идентификатора пользователя с клавиатуры. Данный идентификатор представляет собой последовательность символов a1, a2, ..., aN, где N — количество символов идентификатора (может быть любым), ai — i-й символ идентификатора пользователя.

2. Формирование пароля пользователя b1, b2, ..., bM для данного идентификатора, где M = 6 — количество символов пароля, соответствующее вашему варианту и вывод его на экран. Алгоритм получения символов пароля bi:

b1, b2 — случайные заглавные буквы английского алфавита; b3 = N2 mod10 (где mod 10 — остаток от деления числа на 10); b4 — случайная цифра; b5 — случайный символ из множества {!, ”, #, $, %, &, ’, (, ), \*}; b6 — случайная малая буква английского алфавита.

**Листинг программы:**

**#программа составлена на языке python3**

**#(https://www.python.org/downloads/release/python-390/)**

**#в ходе создания программы был использован**

**#генератор форм page (http://page.sourceforge.net/)**

**import tkinter as tk**

**import tkinter.ttk as ttk**

**#tkinter - библиотека для работы с оконными приложениями**

**import random**

**#функция блокирует ввод с клавиатуры, кроме ctrl+C (копирование)**

**def ctrlEvent(e):**

**if (e.state == 20 and e.keysym == 'c'): return**

**return "break" #заблокировать ввод**

**#**

**abc = 'abcdefghijklmnopqrstuvwxyz'**

**ABC = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'**

**num = '0123456789'**

**symbol = '!”#$%&’,\*'**

**# функция, генерирующая пароль по заданному алгоритму**

**def generate(self):**

**#random.choice(a) - выбрать случайный элемент из списка "a"**

**b1=random.choice(ABC)**

**b2=random.choice(ABC)**

**b3=str(len(self.Text1.get()) % 10)**

**b4=random.choice(num)**

**b5=random.choice(symbol)**

**b6=random.choice(abc)**

**self.Text2.delete('0', 'end') #**

**self.Text2.insert('0', b1+b2+b3+b4+b5+b6) # вывод результата**

**#**

**class Form:**

**def \_\_init\_\_(self, top=None):**

**\_bgcolor = '#d9d9d9' # X11 color: 'gray85'**

**\_fgcolor = '#000000' # X11 color: 'black'**

**\_compcolor = '#d9d9d9' # X11 color: 'gray85'**

**\_ana1color = '#d9d9d9' # X11 color: 'gray85'**

**\_ana2color = '#ececec' # Closest X11 color: 'gray92'**

**top.geometry("520x115+1041+208")**

**top.minsize(1, 1)**

**top.maxsize(1905, 1050)**

**top.resizable(1, 1)**

**top.title("")**

**self.toplevel=top**

**self.Frame1 = tk.Frame(top)**

**self.Frame1.place(relx=0.019, rely=0.087, relheight=0.817**

**, relwidth=0.954)**

**self.Frame1.configure(relief='groove')**

**self.Frame1.configure(borderwidth="2")**

**self.Frame1.configure(relief="groove")**

**self.Button1 = tk.Button(self.Frame1)**

**self.Button1.place(relx=0.02, rely=0.553, height=32, width=185)**

**self.Button1.configure(activebackground="#f9f9f9")**

**self.Button1.configure(command=lambda: generate(self))**

**self.Button1.configure(font="-family {Liberation Sans} -size 12 -weight normal -slant roman -underline 0 -overstrike 0")**

**self.Button1.configure(text='''Сгенерировать пароль''')**

**self.Text1 = tk.Entry(self.Frame1)**

**self.Text1.place(relx=0.423, rely=0.106, relheight=0.298, relwidth=0.552)**

**self.Text1.configure(background="white")**

**self.Text1.configure(font="-family {Liberation Sans} -size 12")**

**self.Text1.configure(selectbackground="blue")**

**self.Text1.configure(selectforeground="white")**

**self.Label1 = tk.Label(self.Frame1)**

**self.Label1.place(relx=0.02, rely=0.106, height=29, width=194)**

**self.Label1.configure(activebackground="#f9f9f9")**

**self.Label1.configure(font="-family {Liberation Sans} -size 12 -weight normal -slant roman -underline 0 -overstrike 0")**

**self.Label1.configure(text='''�дентификатор''')**

**self.Text2 = tk.Entry(self.Frame1)**

**self.Text2.place(relx=0.423, rely=0.574, relheight=0.298**

**, relwidth=0.552)**

**self.Text2.bind("<Key>", lambda e: ctrlEvent(e))#отследить ввод с клавиатуры в**

**# текстовом поле 2**

**self.Text2.configure(background="white")**

**self.Text2.configure(font="-family {Liberation Sans} -size 12")**

**self.Text2.configure(selectbackground="blue")**

**self.Text2.configure(selectforeground="white")**

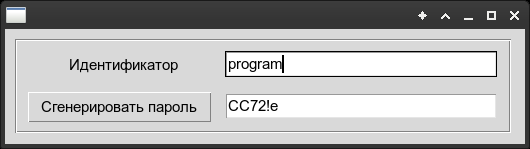
**if \_\_name\_\_ == '\_\_main\_\_':**

**root=tk.Tk()**

**w=Form(root)**

**w.toplevel.mainloop()**

**Результат выполнения программы:**

****

**Выводы:** я изучил основные теоретические сведения и получил практические навыки по оценке стойкости парольной защиты.

**Контрольные вопросы**

1. Дать определение стойкости пароля к взлому. Написать формулу.

2. Дать определение мощности алфавита паролей.

3. Перечислить основные задачи, которые могут решаться с использованием определения стойкости пароля.

4. Перечислить основные требования к выбору пароля.

**Ответы**

1. Стойкость пароля к взлому — это вероятность подбора пароля злоумышленником в течении срока его действия, вычисляется по формуле:

, где A — мощность алфавита паролей, L — длина пароля, S = AL — число всевозможных паролей длины L, которые можно составить из символов алфавита A, V — скорость перебора паролей злоумышленником, T — максимальный срок действия пароля.

2. Мощность алфавита паролей - количество символов, которые могут быть использованы при составлении пароля

3. Основные задачи, которые могут решаться с использованием определения стойкости пароля — это проектирование и реализация программного обеспечения систем аутентификации.

4. Основные требования к выбору пароля:

Длина пароля (количество символов в пароле ) не меньше минимальной длины.

Пароль не должен содержать трех и более одинаковых символов подряд.

Пароль не должен содержать общеупотребительные слова, имена, названия предметов.

Пароль не должен содержать последовательности, пароль должен иметь уникальную (случайную) комбинацию символов