

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Разработка Telegram бота для генерации
паролей

Разработал

Грешнихин Д.В.

2022

1 Цели проекта

В наше время информация играет важную роль в мире. Не менее важную роль играет защита информации, призванная «скрыть от ненужных глаз» важные для людей данные. Один из способов защиты информации – паролирование. Паролирование сейчас применяется повсеместно. Чаще всего паролирование встречается в Интернете. Однако и в жизни часто встречаются пароли: например, банкомат требует ПИН-код, который тоже является паролем. Такой же код требуется ввести при включении телефона.

Пароль – это секретное слово или набор символов, предназначенный для подтверждения личности или полномочий. Пароли часто используются для защиты информации от несанкционированного доступа. В большинстве вычислительных систем комбинация «имя пользователя - пароль» используется для удостоверения пользователя. Паролирование - основной способ защиты информации.

Зачастую пользователи создают пароли на основе персональных данных, каких-либо памятных событий, имён друзей, животных, что является небезопасным решением. Поэтому, так важно использовать программные продукты для генерации таких паролей, создание которых не от чего не зависят.

Таким образом, целью проекта является создание Telegram бота, выполняющий функцию генератора паролей, а также вывод советов по информационной безопасности.

Telegram – кроссплатформенная система мгновенного обмена сообщениями (мессенджер) с функциями VoIP, позволяющая обмениваться текстовыми, голосовыми и видеосообщениями, стикерами и фотографиями, файлами многих форматов. Также можно совершать видео и аудиозвонки и трансляции в каналах и группах, организовывать конференции, многопользовательские группы и каналы. С помощью ботов функционал приложения практически не ограничен. Клиентские приложения Telegram доступны для Android, iOS, Windows, macOS и GNU/Linux.

Бот — специальная программа, выполняющая автоматически и/или по заданному расписанию какие-либо действия через интерфейсы, предназначенные для людей.

Обычно боты предназначаются для выполнения работы, однообразной и повторяемой, с максимально возможной скоростью (очевидно, намного выше возможностей человека). Лицо, обслуживающее серверы, может поместить на сервере файл robots.txt, содержащий ограничения, которым обязаны подчиняться боты.

Боты находят также применение в условиях, когда требуется лучшая реакция по сравнению с возможностями человека (например, игровые боты, боты для интернет-аукционов и тому подобное) или, реже, для имитации действий человека (например, боты для чатов и тому подобное).

2 Руководство пользователя

Для начала работы с ботом нужно ввести в поисковик Telegram'a ввести «MartellPasswordGenerator_bot» или перейти по ссылке.¹ После открытия диалогового окна можно приступить к работе с программным продуктом. Диалоговое окно представлено на рисунке 1

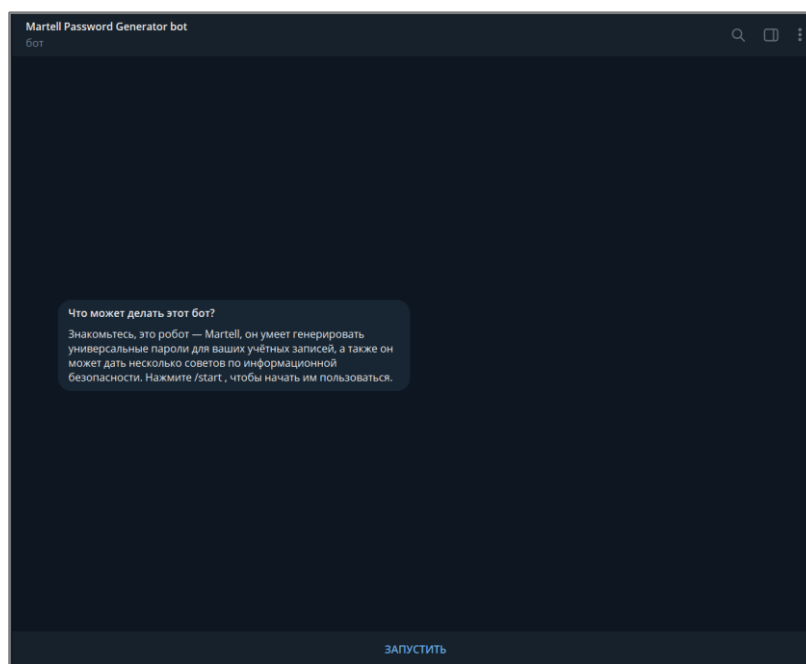


Рисунок 1 – Диалоговое окно

Чтобы начать работу с программным продуктом необходимо нажать кнопку «Запустить». Данный процесс изображён на рисунке 2

¹ Ссылка на Telegram бота: https://t.me/MartellPasswordGenerator_bot

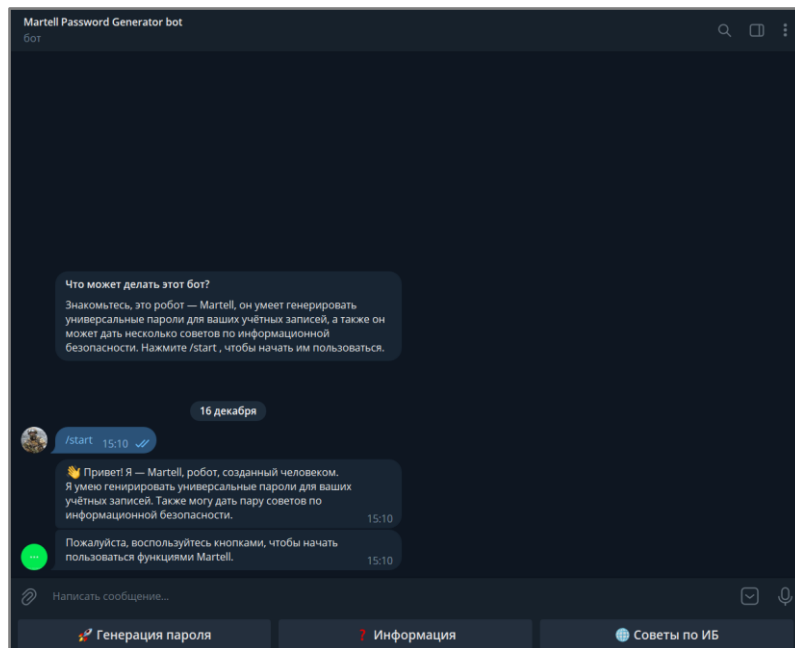


Рисунок 2 – Запуск бота

Основной функцией бота является генерация паролей. Для этого нужно нажать кнопку «Генерация пароля». Процесс его создания представлен на рисунке 3

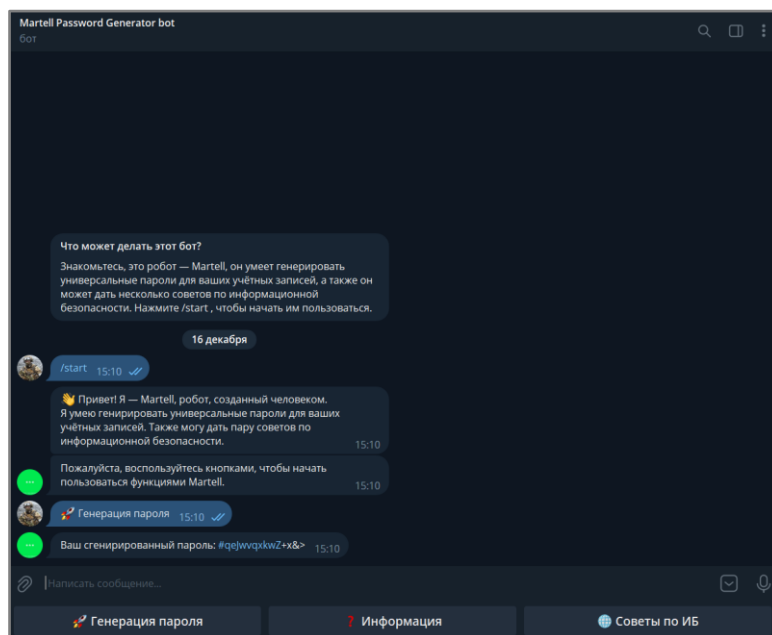


Рисунок 3 – Генерация пароля

Следующей функцией является вывод советов по информационной безопасности. Для отображения совета необходимо нажать на кнопку «советы по ИБ». Их вывод представлен на рисунке 4

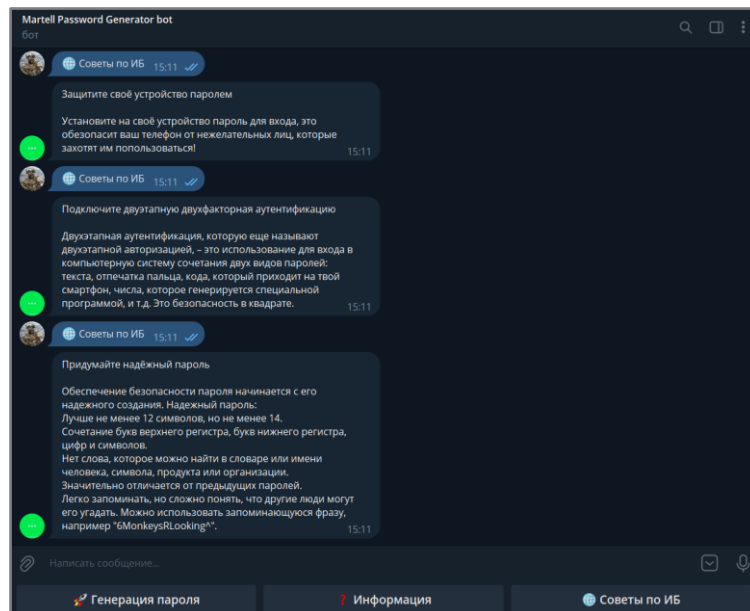


Рисунок 4 – Вывод советом по ИБ

Также по нажатии на кнопку «Информация» можно узнать информацию о боте, что представлено на рисунке 5

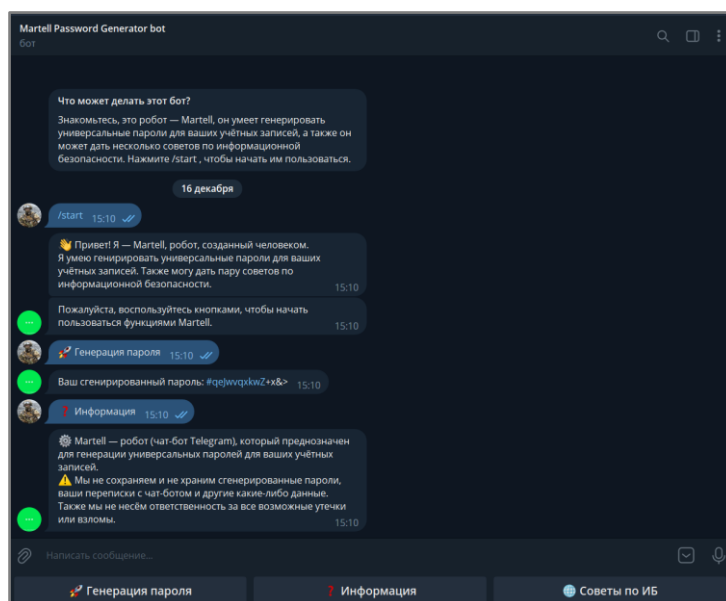


Рисунок 5 – Вывод информации о боте

3 Инструменты для реализации продукта

В качестве языка программирования для разработки бота выбран язык Python.

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Плюсы языка программирования состоят в следующем:

- 1) хорошо подходит для новичков;
- 2) простой минималистичный синтаксис: код легко писать, читать и поддерживать;
- 3) большая стандартная библиотека и много дополнительных библиотек;
- 4) большой выбор фреймворков;
- 5) поддерживает объектно-ориентированное программирование и другие парадигмы;
- 6) кроссплатформенность и поддержка почти всех современных систем.

Минусы языка программирования состоят в следующем:

- 1) низкая скорость;
- 2) плохо подходит для разработки мобильных приложений;
- 3) из-за динамической типизации выше вероятность ошибки при запуске, нужно больше тестов;
- 4) не подходит для работы с памятью на низком уровне.

В качестве среды разработки выбрано Visual Studio.

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ,

поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Плюсы среды разработки состоят в следующем:

- 1) поддержка C#, VB, XAML, HTML, JavaScript, TS и других языков;
- 2) отлично подходит для создания различного программного обеспечения: ASP.Net, Xamarin и так далее;
- 3) мощная поддержка навигации и рефакторинга;
- 4) прекрасно реализованная поддержка интеллектуальных сочетаний клавиш;
- 5) интеграция с Visual Studio и Unity.

Минусы среды разработки состоят в следующем:

- 1) версии Visual Studio Professional и Visual Studio Enterprise являются платными;
- 2) тяжеловесная IDE, достаточно требовательная к железу;
- 3) ряд разработчиков считают ее избыточной для создания мелких проектов;
- 4) достаточно маленькая скорость запуска проектов и приложений;
- 5) не работает на Linux;
- 6) платные версии с расширенным функционалом распространяются по подписке: Professional – от 45\$ в месяц, Enterprise – от 250\$ в месяц.

Библиотеки, которые были использованы в разработке проекта:

- 1) pyTelegramBotAPI – это синхронная и асинхронная реализация Telegram Bot API.
- 2) Random – это класс, реализующий генератор псевдослучайных чисел по умолчанию, используемый модулем случайных чисел.

4 Структура проекта и описание команд бота

В файле config.py прописан токен для работы с ботом. Файл изображён на рисунке 6

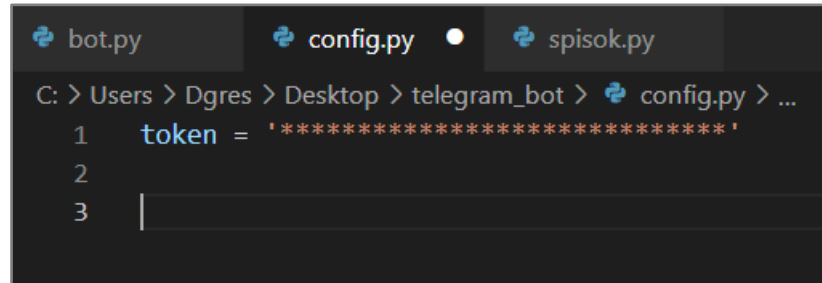


Рисунок 6 – Файл с токеном

В файле spisok.py прописан словарь, содержащий информацию, которая предназначена для работы вывода советов по ИБ. Файл изображён на рисунке 7

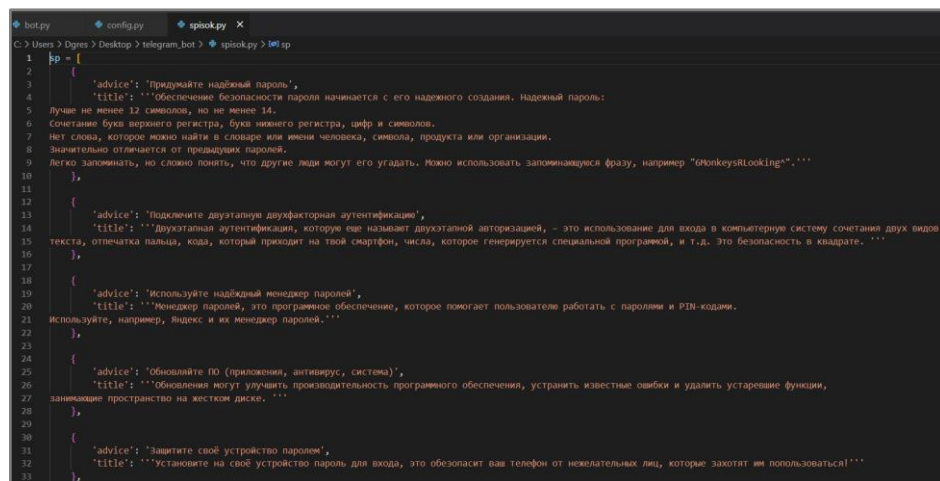


Рисунок 7 – Файл с словарём

В файле bot.py прописан код программного продукта. В начале выполняется процесс импорта библиотек и файлов (config.py и spisok.py). Далее объявляется переменная, которая содержит символы для генерации пароля. Следующим шагом создаётся переменная, которая создаёт бота и связывает его токеном (из файла config.py берём переменную token). Затем прописывается обработчик команды (если вызвана команда /start, то объявляется новая функция, которая выводит приветственное сообщение пользователю и предоставляет три кнопки на выбор:

генерация пароля, информация и советы по ИБ). Далее идёт процесс проверки отправки сообщения пользователя боту, который вызывает функцию обработчика нажатий. Существует три следующих сценария:

- 1) нажата кнопка генерация пароля (объявляется переменная password для записи пароля, далее объявляется цикл, который берёт 15 случайных значений из переменной chars, записывает в переменную password и выводит пользователю в виде сообщения в Telegram);
- 2) нажата кнопка информация (выводит пользователю информацию в виде сообщения в Telegram);
- 3) нажата кнопка советы по ИБ (из файла spisok.py, из словаря берётся случайный совет и описание по функции random.choice и записывается в другие две переменные, которые «раскрывают» словарь и отправляется сообщение пользователю).

В конце прописана функция, которая означает бесконечно выполняющийся цикл запросов к серверам Telegram.

Программный код класса bot.py представлен на рисунках 8-9

```
import telebot
import config
from telebot import types
import spisok
import random

chars = '+-/*!&$%?~<>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'

bot = telebot.TeleBot(config.token)

@bot.message_handler(commands=['start'])
def start(message):
    bot.send_message(message.chat.id, text='💖 Привет! Я – Martell, робот, созданный человеком. Я умею генерировать универсальные пароли для ваших учётных записей. Также могу дать пару советов по информационной безопасности.')
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    btn1 = types.KeyboardButton("🔑 Генерация пароля")
    btn2 = types.KeyboardButton("❓ Информация")
    btn3 = types.KeyboardButton("💡 Советы по ИБ")
    markup.add(btn1, btn2, btn3)
    bot.send_message(message.chat.id, text='Пожалуйста, воспользуйтесь кнопками, чтобы начать пользоваться функциями Martell.', reply_markup=markup)

@bot.message_handler(content_types=['text'])
def buttons(message):
    if message.text == "🔑 Генерация пароля":
        password = ''
        for i in range(15):
            password += random.choice(chars)
        bot.send_message(message.chat.id, text=f'Ваш сгенерированный пароль: {password}')

    elif message.text == "❓ Информация":
        bot.send_message(message.chat.id, text=f'💡 Martell – робот, который предназначен для генерации универсальных паролей для ваших учётных записей. Мы не сохраняем и не храним сгенерированные пароли, ваши переписки с чат-ботом и другие какие-либо данные. Также мы не несём ответственность за все возможные утечки и
```

Рисунок 8 – Программный код

```

@bot.message_handler(content_types=['text'])
def buttons(message):
    if(message.text == "🔑 Генерация пароля"):
        password = ''
        for i in range(15):
            password += random.choice(chars)
        bot.send_message(message.chat.id, text=f'''Ваш сгенерированный пароль: {password}''')

    elif(message.text == "❓ Информация"):
        bot.send_message(message.chat.id, text=f'''🤖 Martell – робот, который предназначен для генерации универсальных паролей для ваших учётных записей.
        ⚠️ Мы не сохраняем и не храним сгенерированные пароли, ваши переписки с чат-ботом и другие какие-либо данные. Также мы не несём ответственность за все возможные утечки и

    elif(message.text == "💡 Советы по ИБ"):
        b = random.choice(spisok.sp)
        b1 = b['advice']
        b2 = b['title']
        bot.send_message(message.chat.id, text=f'''{b1}\n
        {b2}''')

bot.polling(none_stop=True)

```

Рисунок 9 – Программный код