|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе № 1**

**Вариант № 9**

**Название:** реализация простейшего генератора паролей

**Дисциплина:** Информационная безопасность автоматизированных систем

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-31М |  |  | И.С. Марчук |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Д.А. Миков |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2024

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

**Задание:** 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 — количество символов пароля, соответствующее вашему варианту и вывод его на экран. Алгоритм получения символов пароля bi указан в перечне требований для вашего варианта (таблица 1).

Вариант: 9;

М: 12;

Перечень требований:

* b1, ..., b1 + Q — случайные малые буквы русского алфавита, где Q =N 3mod5;
* b1+Q+1, ..., b1+Q+1+P — случайные заглавные буквы русского алфавита, где P = N 2mod6;
* Оставшиеся символы пароля — случайные цифры.

**Ход работы**

Стойкость к взлому подсистемы парольной идентификации (аутентификации) во многом определяется тем, насколько правильно были сформированы пароли пользователей. При несоблюдении ряда требований к выбору паролей, данная стойкость в значительной степени уменьшается, и подсистема идентификации (аутентификации) становится достаточно уязвима при правильно построенной атаке.

Ниже перечислены основные требования, которые должны быть учтены при выборе пароля пользователя.

1. Минимальная длина пароля должна быть не менее 6 символов. Сокращение длины пароля во многом повышает вероятность успешной атаки полным их перебором.

2. Пароль должен состоять из различных групп символов (малые и большие латинские буквы, цифры, специальные символы «(», «)», «#» и т. д.). Использование одной конкретной группы символов при формировании пароля в значительной степени повышает вероятность успешной атаки по маске.

3. В качестве пароля не должны использоваться реальные слова, имена, фамилии и т. д. Использование в качестве паролей конкретных слов, имен в значительной степени повышает вероятность успешной атаки по словарю.

Иногда генераторы паролей могут использовать при данном генерировании элементы, входящие в идентификатор пользователя (отдельные его символы, количество символов и т. д.). В отдельных вариантах пароль может формироваться даже целиком из идентификатора на основе некоторого алгоритма. В последнем случае заданному идентификатору пользователя ставится в соответствие единственный пароль, который формируется на основе идентификатора.

**Написание программы**

Я реализовал на языке Kotlin консольную программу, которая позволила генерировать пароли в соответствии с представленным перечнем требований. Исходный код программы представлен в листинге 1.

Листинг программы 1 – Программа генерации пароля

package org.example;

import java.util.Random;

import java.util.Scanner;

// Press Shift twice to open the Search Everywhere dialog and type `show whitespaces`,

// then press Enter. You can now see whitespace characters in your code.

public class Main {

static Scanner scanner = new Scanner(System.in);

static Random randomizer = new Random(System.currentTimeMillis());

public static void main(String[] args) {

System.out.println("Генератор паролей. Марчук Иван ИУ6-31М ЛР1");

System.out.print("Введите имя пользователя:");

String input = scanner.nextLine();

System.out.println("----- Вычисляем -----");

int mParam = 12;

int nParam = input.length();

int qParam = (nParam \* nParam \* nParam) % 5;

int pParam = (nParam \* nParam) % 6;

System.out.println("M=" + mParam);

System.out.println("N=" + nParam);

System.out.println("Q=" + qParam);

System.out.println("P=" + pParam);

// генерация строки пароля

int charPosition = 1;

char[] password = new char[mParam];

System.out.println("Генерация b1, ..., b(1 + Q=" + qParam + ")");

for (; charPosition <= qParam + 1; charPosition++) {

password[charPosition-1] = (char) (1072 + randomizer.nextInt(31));

}

System.out.println(password);

System.out.println("Генерация b(1+Q=" + qParam +

"+1), ..., b(1+Q=" + qParam + "+1+P=" + pParam + ")");

for (; charPosition <= 1 + qParam + 1 + pParam; charPosition++) {

password[charPosition-1] = (char) (1040 + randomizer.nextInt(31));

}

System.out.println(password);

System.out.println("Генерация цифр ");

for (; charPosition <= mParam; charPosition++) {

password[charPosition-1] = (char) (48 + randomizer.nextInt(10));

}

System.out.print("-------------------------\nСгенерированный пароль: ");

System.out.println(password);

}

}

**Пример работы программы**

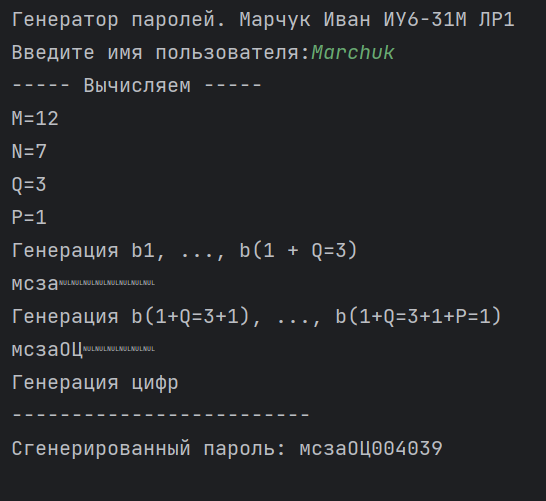


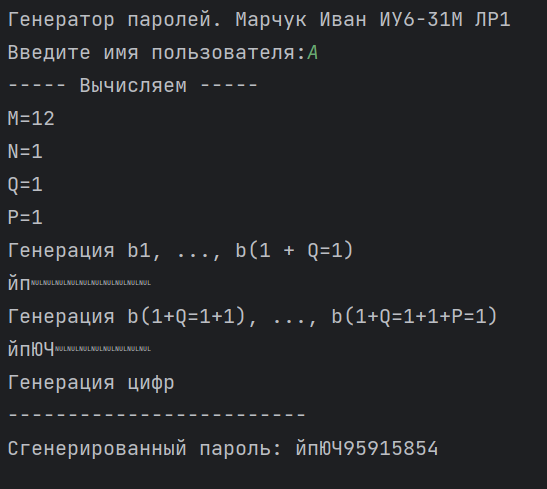
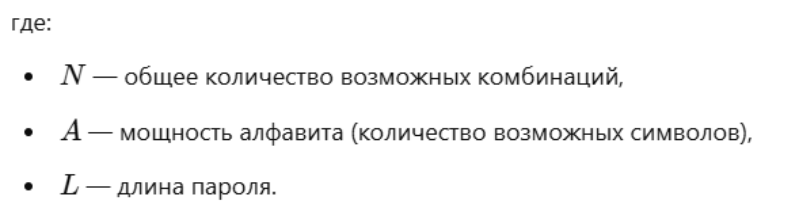
Рисунок 1 – Пример работы программы 

Рисунок 2 – Пример работы программы

**Контрольные вопросы**

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

Стойкость пароля к взлому — это мера, характеризующая, насколько сложно злоумышленнику угадать или подобрать пароль с использованием различных методов взлома, таких как атаки по словарю, грубая сила (brute force) и другие. Стойкость пароля выражается через количество возможных комбинаций пароля, которое злоумышленнику нужно перебрать. Формула для оценки стойкости пароля:



2. Дать определение мощности алфавита паролей.

Мощность алфавита паролей — это количество различных символов, которые могут быть использованы для создания пароля. Например, если для создания пароля можно использовать только цифры (0-9), мощность алфавита составляет 10, если можно использовать строчные и прописные латинские буквы (A-Z, a-z) и цифры, то мощность алфавита равна 62.

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

- Оценка уровня безопасности паролей.

- Подбор рекомендуемых требований к паролям для обеспечения должного уровня защиты.

- Оценка времени, необходимого для взлома пароля с использованием атаки грубой силы.

- Анализ рисков, связанных с использованием слабых паролей.

- Разработка политики смены паролей в зависимости от их стойкости.

4. Перечислить основные требования к выбору пароля.

- Длина пароля должна быть не менее 8-12 символов.

- Пароль должен включать символы разного типа: заглавные и строчные буквы, цифры, специальные символы (например, !, @, #, $).

- Пароль не должен содержать легко угадываемую информацию, такую как имена, даты рождения или популярные слова.

- Пароль не должен повторять ранее использованные пароли.

- Рекомендуется использовать уникальные пароли для разных учетных записей.

- Использование менеджеров паролей для создания и хранения сложных уникальных паролей.

**Вывод**

Я реализовал простейший генератор паролей, обладающий основными требованиями к парольным генераторам. Я получил основные теоретические сведения и практические навыки по оценке стойкости парольной защиты.