Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 1

Тема «Анализ надёжности пароля»

по дисциплине «Информационная безопасность и защита информации»

Выполнил: студент группы ИСТ-22-1б Петраков М.В.

Проверил: доцент каф. ВММБ Труфанов А. Н.

Пермь, 2025

Цель работы

1. Ознакомиться с основными методами криптографической защиты информации.
2. Получить практические навыки создания программного модуля оценки надежности выбранного пароля.

1. Описание алгоритма надежности пароля

Объявляются булевы переменные существования символов ASCII, цифр, заглавных и строчных символов. Далее происходит перебор всех символов в пароле, где определяется, из каких групп символов может состоять пароль. В зависимости от существования символов различных групп формируется размер алфавита (10, 26, 36, 52, 62, 95). После определяется мощность пространства паролей и устанавливаются следующие параметры: скорость перебора паролей в секунду, количество неверных попыток, после достижения которых включается пауза на перебор, а также время паузы.

Затем определяется время перебора паролей. Объявлено, что это время измеряется минимум в секундах. Исходя из этого ограничения устанавливается такое условие: если время перебора имеет дробную часть, не равную нулю, это время округляется до следующего целого числа. Необходимо также учитывать отношение (целая часть от деления) мощности пространства и количества неверных попыток, после которых идёт пауза. В случае деления нацело последняя пауза не должна существовать (количество пауз должно быть уменьшено на 1). В противном случае количество пауз изменять не нужно.

Последним действием является форматирование количества времени. Время в секундах переводится во время в годах, месяцах, днях, часах, минутах и секундах.

2. Результаты измерения времени перебора паролей различных длин и алфавитов

Далее представлена таблица для паролей длины 6-8 символов алфавитов мощностью 10, 26, 36, 95 при отсутствии паузы.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Мощность алфавита | Длина пароля | Количество комбинаций |
| 1 | 10 | 6 | 1000000 |
| 2 | 26 | 6 | 308915776 |
| 3 | 36 | 6 | 2176782336 |
| 4 | 95 | 6 | 735091890625 |
| 5 | 10 | 7 | 10000000 |
| 6 | 26 | 7 | 8031810176 |
| 7 | 36 | 7 | 78364164096 |
| 8 | 95 | 7 | 69833729609375 |
| 9 | 10 | 8 | 100000000 |
| 10 | 26 | 8 | 208827064576 |
| 11 | 36 | 8 | 2821109907456 |
| 12 | 95 | 8 | 6634204312890625 |

Приложение. Код Python

def passwords(password):  
 ascii\_exist = False  
 digits\_exist = False  
 low\_letters\_exist = False  
 big\_letters\_exist = False  
  
 alphabet = 0  
  
 # проверить каждый символ на принадлежность к какому-либо алфавиту  
 for letter in password:  
 if str(letter).isdigit():  
 digits\_exist = True  
 if str(letter).islower():  
 low\_letters\_exist = True  
 if str(letter).isupper():  
 big\_letters\_exist = True  
 if (str(letter).isascii() and not (str(letter).isupper()) and  
 not (str(letter).islower()) and not (str(letter).isdigit())):  
 ascii\_exist = True  
 # и в зависимости от принадлежности добавить к объявленному алфавиту мощность присутствующего  
 if digits\_exist: alphabet += 10  
 if big\_letters\_exist: alphabet += 26  
 if low\_letters\_exist: alphabet += 26  
 if ascii\_exist: alphabet = 95  
  
 print(f"Алфавит равен {alphabet}")  
  
 power = alphabet \*\* (len(password)) # количество всевозможных комбинаций  
  
 print(f"Мощность пространства равна {power}")  
  
 s = 1000 # скорость перебора паролей в секунду  
 m = 5 # количество неправильных попыток, после которых идёт пауза в v секунд  
 v = 0 # пауза в секундах при неверном вводе пароля  
  
 if power % m == 0:  
 times\_to\_pause = power // m - 1 # сколько раз будет спровоцирована пауза  
 else:  
 times\_to\_pause = power // m  
 t = (power // s + 1) + times\_to\_pause \* v  
  
 # обнаружен косяк - если мощность делится нацело на количество неверных попыток  
 # то будет лишняя пауза  
  
 return t  
  
  
def reformat\_time(time):  
 seconds = time % 60  
 all\_minutes = time // 60  
 minutes = all\_minutes % 60  
 all\_hours = all\_minutes // 60  
 hours = all\_hours % 24  
 all\_days = all\_hours // 24  
 days = all\_days % 30  
 all\_months = all\_days // 30  
 months = all\_months % 12  
 years = all\_days // 365  
 print(f"{years} лет, {months} месяцев, {days} дней, {hours} часов, {minutes} минут, {seconds} секунд")  
 pass  
  
  
print("Введите пароль")  
password = input()  
reformat\_time(passwords(password))