МАТЕРИАЛЫ ЗАДАНИЙ МЕЖРЕГИОНАЛЬНОЙ ОЛИМПИАДЫ ШКОЛЬНИКОВ ИМЕНИ И.Я. ВЕРЧЕНКО ПО ПРОФИЛЮ «КОМПЬЮТЕРНАЯ БЕЗОПАСНОСТЬ» (2024/2025 УЧЕБНЫЙ ГОД)

ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП
11 КЛАСС

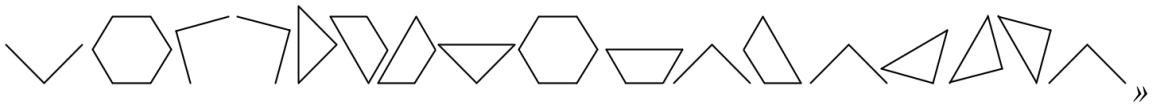
СОДЕРЖАНИЕ

Задача 1. Солнечный шифр

Вариант 1

Отдел информационной безопасности перехватил подозрительные письма от одного из сотрудников компании на подозрительный почтовый ящик. Письма всегда содержали время, странную последовательность символов и вложение с геометрической фигурой. Вот последнее сообщение, которое удалось перехватить:

«В – 12 часов, и – 25-35 минут, буду ждать тебя



В ходе анализа отдел информационной безопасности не смог расшифровать часть, где встречались странные символы. Было установлено, что это побитая кодировка, однако после этих сообщений были зафиксированы утечки конфиденциальной информации. Также, во вложении к письму была прикреплена картинка со странной геометрической фигурой и расположенными на ней буквами (рисунок 1).

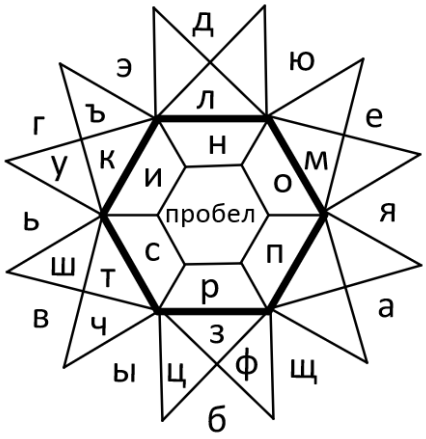


Рисунок 1 – Странная геометрическая фигура

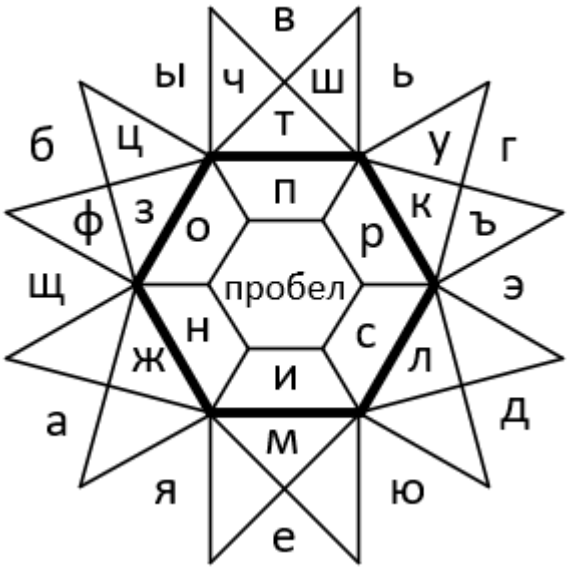
Отдел информационной безопасности считает, что с помощью этой фигуры и предоставленного письма можно расшифровать сообщение. Ваша задача на основе предоставленных данных установить место очередной передачи конфиденциальной информации. В ответе укажите исходное сообщение.

Решение

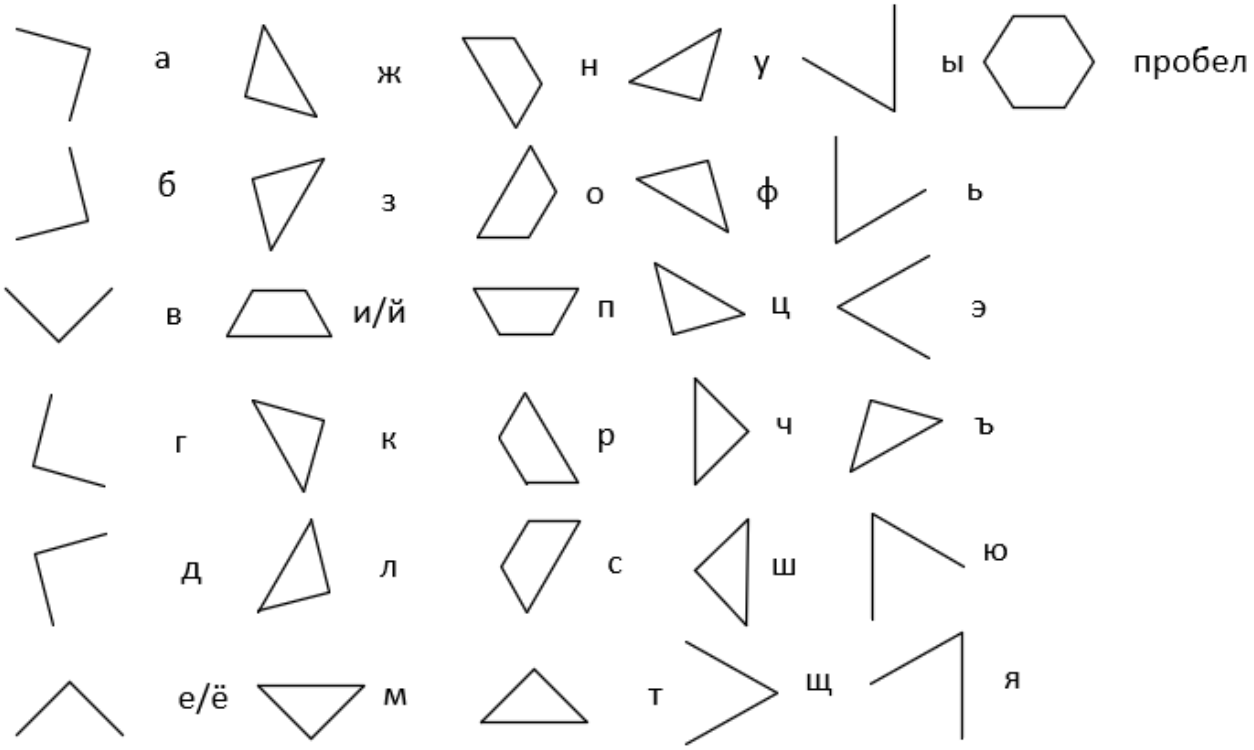
Для начала проанализируем представленные письмо и вложение и поймем, что все буквы соответствуют какой-либо фигуре внутри одной большой фигуры. Если начинать расшифровывать «в лоб», то ничего осмысленного мы не получаем:

«д авцсиз нбоб--мб».

Если вчитаться в условие задачи, то нам известно, что сотрудник каждый раз присылал время и вложение. Отсюда можно сделать вывод, что время и вложение как-то взаимосвязаны. Количество букв по внешнему периметру геометрической фигуры равно 12, что совпадает с количеством часов на циферблате часов. Количество букв во внутреннем шестиугольнике – 6, что равняется количеству промежутков по 10 минут в одном часе. Из этого получается, что внешний круг необходимо повернуть так, чтобы буква «в» стояла на 12 часов, т.к. в письме указано: «в – 12 часов», а внутренний шестиугольник нужно повернуть так, чтобы буква «и» стояла в промежутке 25-35 минут, т.к. в письме указано: «и – 25-35 минут». Искомая фигура будет выглядеть следующим образом:



Если разложить фигуру по мелким фигурам, то получим следующий алфавит:



Теперь можно приступить к расшифровке. Место, где будут передаваться конфиденциальные данные - «в дачном переулке».

Ответ: в дачном переулке

Вариант 2

Отдел информационной безопасности перехватил подозрительные письма от одного из сотрудников компании на подозрительный почтовый ящик. Письма всегда содержали время, странную последовательность символов и вложение с геометрической фигурой. Вот последнее сообщение, которое удалось перехватить:

«В – 6 часов, и – 5-15 минут, буду ждать тебя

»

В ходе анализа отдел информационной безопасности не смог расшифровать часть, где встречались странные символы. Было установлено, что это побитая кодировка, однако после этих сообщений были зафиксированы утечки конфиденциальной информации. Также, во вложении к письму была прикреплена картинка со странной геометрической фигурой и расположенными на ней буквами (рисунок 1).

.

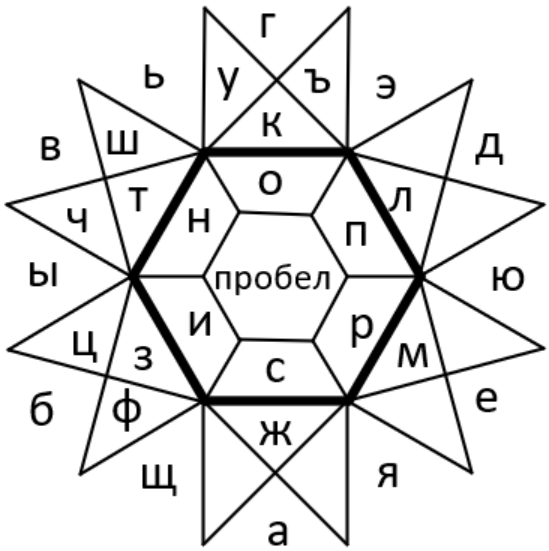


Рисунок 1 – Странная геометрическая фигура

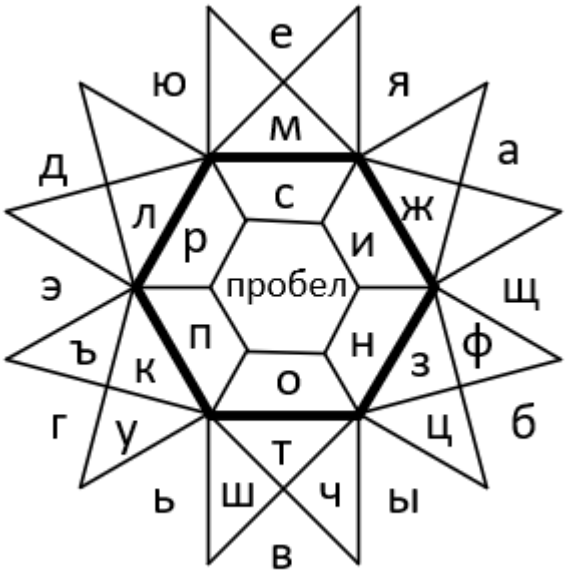
Отдел информационной безопасности считает, что с помощью этой фигуры и предоставленного письма можно расшифровать сообщение. Ваша задача на основе предоставленных данных установить место очередной передачи конфиденциальной информации. В ответе укажите исходное сообщение.

Решение

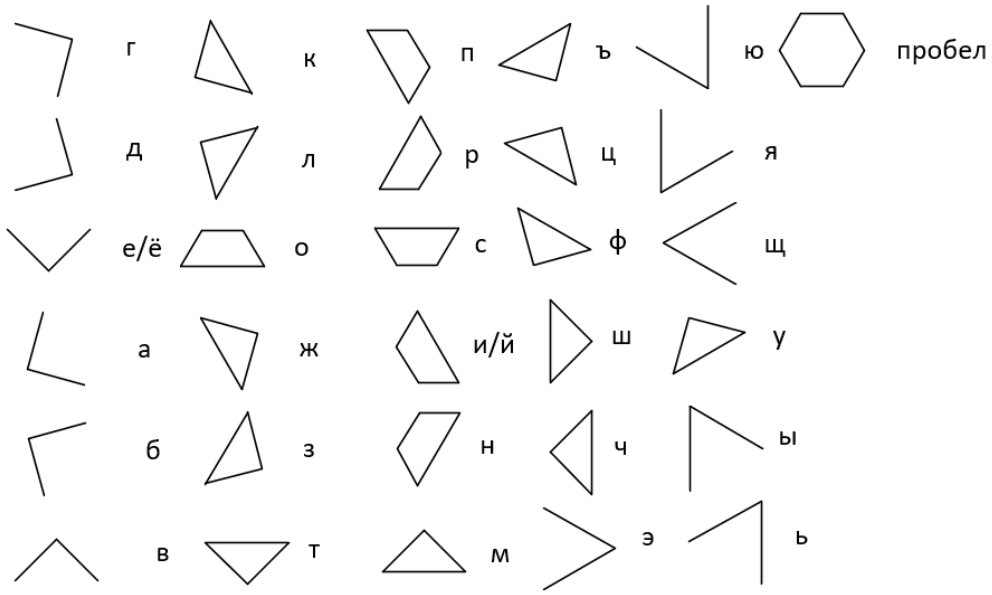
Для начала проанализируем представленные письмо и вложение и поймем, что все буквы соответствуют какой-либо фигуре внутри одной большой фигуры. Если начинать расшифровывать «в лоб», то ничего осмысленного мы не получаем:

«а здшг ужнфвгтщ».

Если вчитаться в условие задачи, то нам известно, что сотрудник каждый раз присылал время и вложение. Отсюда можно сделать вывод, что время и вложение как-то взаимосвязаны. Количество букв по внешнему периметру геометрической фигуры равно 12, что совпадает с количеством часов на циферблате часов. Количество букв во внутреннем шестиугольнике – 6, что равняется количеству промежутков по 10 минут в одном часе. Из этого получается, что внешний круг необходимо повернуть так, чтобы буква «в» стояла на 6 часов, т.к. в письме указано: «в – 6 часов», а внутренний шестиугольник нужно повернуть так, чтобы буква «и» стояла в промежутке 5-15 минут, т.к. в письме указано: «и – 5-15 минут». Искомая фигура будет выглядеть следующим образом:



Если разложить фигуру по мелким фигурам, то получим следующий алфавит:



Теперь можно приступить к расшифровке. Место, где будут передаваться конфиденциальные данные - «в кафе штрудель».

Ответ: в кафе штрудель

Задача 2. Таблица символов

Вариант 1

В предполагаемом месте встречи агента был найден ноутбук. В ходе анализа его содержимого была обнаружена таблица символов (таблица 1) и скриншот формулы (рисунок 1).

Экспертами установлено, что в таблице символов скрыт пароль от рабочего стола ноутбука, каждая буква которого находится на определенной позиции, вычисляемой с помощью указанной формулы.

Определите пароль, если известно, что координаты первого символа: $(x_0, y_0) = (5, 6)$, длина пароля – *13* символов, и пароль является осмысленным словом. В ответе укажите сам пароль и значения параметров формулы.

Таблица 1 – Таблица символов

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О
1	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	О	Ю	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	О	Ю
2	Я	А	Б	В	Г	Н	Е	Ё	Ж	Ф	И	Й	К	Л	М	Н	Я	Ц	Б	В	Г	Н	Е	Ё	Ж	З	И	Й	К	Л	М	Н
3	О	П	Р	С	Т	У	Ф	Ч	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	О	П	Р	С	Т	У	Ф	Ч	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
4	Ю	Я	А	Я	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Ю	Я	А	Я	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М
5	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ц	Ш	Щ	Ъ	Ы	Ь	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ц	Ш	Щ	Ъ	Ы	Ь
6	Э	Ю	Я	А	Б	К	Г	Д	Е	Ё	Ж	З	И	С	К	Л	Э	Ю	Я	А	Б	К	Г	Д	Е	Ё	Ж	З	И	Я	К	Л
7	М	Н	О	П	Р	С	Т	У	Н	Х	Ц	Ч	Ш	Щ	Ъ	Ы	М	Н	О	П	Р	С	Т	У	Н	Х	Ц	Ч	Ш	Щ	Ъ	Ы
8	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	Е	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	Е
9	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
10	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	И	И	Й	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	И	И	Й
11	К	Л	М	Н	С	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	К	Л	М	Н	С	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
12	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	О	Ё	Ж	З	И	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	О	Ё	Ж	З	И
13	Й	К	Е	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Й	К	Е	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
14	Щ	А	Ы	Ь	Э	Ю	Я	А	Б	Т	Т	Д	Е	Ё	Ж	З	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	Т	Т	Д	Е	Ё	Ж	З
15	Л	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Л	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
16	А	Б	В	Г	Д	Е	Ё	И	З	И	Й	К	Л	М	Н	С	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	И
17	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	О	Ю	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	О	Ю
18	Я	А	Б	В	Г	Н	Е	Ё	Ж	З	И	Й	К	Л	М	Н	Я	А	Б	В	Г	Н	Е	Ё	Ж	З	И	Й	К	Л	М	Н
19	О	П	Р	С	Т	У	Ф	Ч	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	О	П	Р	С	Т	У	Ф	Ч	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
20	Ю	Я	А	Я	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Ю	Я	А	Я	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М
21	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ц	Ш	Щ	Ъ	Ы	Ь	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ц	Ш	Щ	Ъ	Ы	Ь
22	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л
23	М	Н	О	П	Р	С	Т	У	Н	Х	Ц	Ч	Ш	Щ	Ъ	Ы	М	Н	О	П	Р	С	Т	У	Н	Х	Ц	Ч	Ш	Щ	Ъ	Ы
24	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	Е	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	Е
25	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
26	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	И	И	Й	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	И	И	Й
27	К	Л	М	Н	С	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	К	Л	М	Н	С	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
28	Ъ	Ы	Ь	И	Ю	Я	А	Б	В	Г	Д	Л	Ё	Ж	З	И	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	А	Ё	Ж	З	И
29	Й	К	Е	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Й	К	Е	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
30	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	Т	Т	Д	Е	Ё	Ж	З	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	Т	Т	Д	Е	Ё	Ж	З
31	Л	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Л	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч

$$x_{i+1} = ((x_i + a^2)^{a \mid x_i} + a) \bmod 32,$$
$$y_{i+1} = ((y_i + b^2)^{b \mid y_i} + b) \bmod 32$$

Рисунок 1 – Формулы вычисления координат символов,
mod – операция остатка от деления,
| – побитовое ИЛИ (OR).

Решение

Для решения необходимо подобрать такие a и b, для которых будет вычислен осмысленный пароль. Вручную это делать очень долго. Для этого можем написать программу, которая рассчитает слова длинной 13 символов для всех, a и b по формулам вычисления координат.

```
def solve_password_with_table(x0, y0, letter_table, target_length=13):

  for a in range(32):
    for b in range(32):
      password = ""
      x, y = x0, y0
      coords = []  # Список для хранения координат для каждого символа
      # Выводим первую координату (x0, y0)
      try:
        char = letter_table[y][x]
        password += char
        coords.append((x, y))  # Добавляем первую координату
      except IndexError:
        password = ""
        coords = []
        print(f"a={a} b={b}, Пароль: Ошибка: выход за границы таблицы")
        print("-" * 20)
        continue  # Переходим к следующей комбинации a и b
      # Рассчитываем остальные координаты
      for _ in range(target_length - 1):
        x = ((x + (a ** 2)) ** ((a | x)) + a) % 32
        y = ((y + (b ** 2)) ** ((b | y)) + b) % 32
        try:
          char = letter_table[y][x]
          password += char
          coords.append((x, y))  # Добавляем координаты
        except IndexError:
          password = ""
          coords = []
          break
      print(f"a={a} b={b}, Пароль: {password}")

letter_table = [
  ['А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О',
'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О'],
  ['П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'О', 'Ю',
'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'О', 'Ю'],
  ['Я', 'А', 'Б', 'В', 'Г', 'Н', 'Е', 'Ё', 'Ж', 'Ф', 'И', 'Й', 'К', 'Л', 'М', 'Н',
'Я', 'Ц', 'Б', 'В', 'Г', 'Н', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н'],
  ['О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Ч', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э',
'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Ч', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э'],
  ['Ю', 'Я', 'А', 'Я', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М',
'Ю', 'Я', 'А', 'Я', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М'],
  ['Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ц', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь',
'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ц', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь'],
  ['Э', 'Ю', 'Я', 'А', 'Б', 'К', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'С', 'К', 'Л',
'Э', 'Ю', 'Я', 'А', 'Б', 'К', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Я', 'К', 'Л'],
  ['М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Н', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы',
'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Н', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы'],
  ['Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'Е',
'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'Е'],
  ['Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ',
'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ'],
  ['Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'И', 'И', 'Й',
'Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'И', 'И', 'Й'],
```

```
  ['К', 'Л', 'М', 'Н', 'С', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ',
'К', 'Л', 'М', 'Н', 'С', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ'],
  ['Ъ', 'Ы', 'Ь', 'И', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Л', 'Ё', 'Ж', 'З', 'И',
'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'А', 'Ё', 'Ж', 'З', 'И'],
  ['Й', 'К', 'Е', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш',
'Й', 'К', 'Е', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш'],
  ['Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'Т', 'Т', 'Д', 'Е', 'Ё', 'Ж', 'З',
'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'Т', 'Т', 'Д', 'Е', 'Ё', 'Ж', 'З'],
  ['Л', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч',
'Л', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч'],
  ['А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'И', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'С',
'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'И'],
  ['П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'О', 'Ю',
'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'О', 'Ю'],
  ['Я', 'А', 'Б', 'В', 'Г', 'Н', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н',
'Я', 'А', 'Б', 'В', 'Г', 'Н', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н'],
  ['О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Ч', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э',
'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Ч', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э'],
  ['Ю', 'Я', 'А', 'Я', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М',
'Ю', 'Я', 'А', 'Я', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М'],
  ['Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ц', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь',
'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ц', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь'],
  ['Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л',
'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л'],
  ['М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Н', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы',
'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Н', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы'],
  ['Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'Е',
'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'Е'],
  ['Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ',
'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ'],
  ['Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'И', 'И', 'Й',
'Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'И', 'И', 'Й'],
  ['К', 'Л', 'М', 'Н', 'С', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ',
'К', 'Л', 'М', 'Н', 'С', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ'],
  ['Ъ', 'Ы', 'Ь', 'И', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'Л', 'Ё', 'Ж', 'З', 'И',
'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'В', 'Г', 'Д', 'А', 'Ё', 'Ж', 'З', 'И'],
  ['Й', 'К', 'Е', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш',
'Й', 'К', 'Е', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш'],
  ['Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'Т', 'Т', 'Д', 'Е', 'Ё', 'Ж', 'З',
'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', 'А', 'Б', 'Т', 'Т', 'Д', 'Е', 'Ё', 'Ж', 'З'],
  ['Л', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч',
'Л', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч']
]

# Запускаем решение с заданными параметрами и таблицей
solve_password_with_table(5, 6, letter_table)
```

Видим, что нам подходят a = 18 и b = 13, так как это единственная строчка с осмысленным текстом «КЛАССИФИКАЦИЯ».

Ответ: **КЛАССИФИКАЦИЯ,**
a = 18, b = 13

Вариант 2

В предполагаемом месте встречи агента был найден ноутбук. В ходе анализа его содержимого была обнаружена таблица символов (таблица 1) и скриншот формулы (рисунок 1).

Экспертами установлено, что в таблице символов скрыт пароль от рабочего стола ноутбука, каждая буква которого находится на определенной позиции, вычисляемой с помощью указанной формулы.

Определите пароль, если известно, что координаты первого символа: $(x_0, y_0) = (8, 3)$, длина пароля – *12* символов, и пароль является осмысленным словом. В ответе укажите сам пароль и значения параметров формулы.

Таблица 1 – Таблица символов

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	О	Ф	У	Т	С	Р	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	О	Ф	У	Т	С	Р
1	П	О	Н	М	Л	К	Й	И	З	Ж	Ё	Е	Д	Г	В	Б	П	О	Н	М	Л	К	Я	И	З	Ж	Ё	Е	Д	Г	В	Б
2	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	И	Ч	Ц	Х	Ф	У	Т	С	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	И	Ч	Ц	Х	Ф	У	Т	С
3	Р	П	О	Н	М	Л	К	Й	В	З	Ж	Ё	Е	Д	Г	О	Р	П	О	Н	М	Л	К	Й	И	З	Ж	Ё	Е	Д	Г	О
4	Б	А	Я	Ю	Э	Ь	О	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У	Т	Б	А	Я	Ю	Э	Ь	О	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У	Т
5	С	Р	П	О	Н	М	Л	К	Й	И	З	Ж	Ё	Т	Д	Г	С	Р	П	О	Н	М	Л	К	Й	И	И	Ж	Ё	Т	Д	Г
6	В	Б	А	Я	П	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У	В	Б	А	Я	П	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У
7	Т	С	Р	П	О	Н	М	Л	К	Й	И	Е	Ж	Ё	Е	Д	Т	С	Р	П	З	Н	У	Л	К	Й	И	Е	Ж	Ё	Е	Д
8	Г	В	Я	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф	Г	В	Я	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф
9	У	Т	С	Р	П	О	З	М	Л	С	Й	И	З	Ж	Ё	Е	У	Т	С	Р	П	О	Н	М	Л	С	Й	И	З	Ж	Ё	Е
10	И	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	И	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х
11	Ф	У	Т	С	Р	П	О	Р	М	Л	К	Й	И	З	Ж	Ё	Ф	У	Т	С	Р	П	О	Р	А	Л	К	Й	И	З	Ж	Ё
12	Е	Д	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Е	Д	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц
13	Х	Ф	У	Т	С	Н	П	О	Н	М	Л	К	Й	И	З	Ж	Х	Ф	У	Т	С	Н	П	О	Н	М	Л	К	Й	И	З	Ж
14	Ё	Е	Д	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Т	Щ	Ш	Ч	Ё	Е	Д	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Т	Щ	Ш	Ч
15	Ц	Х	Ф	В	И	С	Р	П	О	Н	М	Л	К	Й	И	З	Ц	Х	Ф	В	Т	С	Р	П	О	Н	М	Л	К	Й	И	З
16	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	О	Ф	У	Т	С	Р	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	О	Ф	У	Т	С	Р
17	П	О	Н	М	Л	К	Й	И	З	Ж	Ё	Е	Д	Г	В	Б	П	О	Н	М	Л	К	Й	И	З	Ж	Ё	Е	Д	Г	В	Б
18	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	И	Ч	Ц	Х	Ф	У	Т	С	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	И	Ч	Ц	Х	Ф	У	Т	С
19	Р	П	О	Н	М	Л	К	Й	А	З	Ж	Ё	Е	Д	Г	О	Р	П	О	Н	М	Л	К	Й	И	З	Ж	Ё	Е	Д	Г	О
20	Б	А	Я	Ю	Э	Ь	О	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У	Т	Б	А	Я	Ю	Э	Ь	О	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У	Т
21	С	Р	П	О	Н	М	Л	К	Й	И	З	Ж	Ё	Т	Д	Г	С	Р	П	О	Н	М	Л	К	Й	И	Ц	Ж	Ё	Т	Д	Г
22	В	Б	А	Я	П	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У	В	Б	А	Я	П	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У
23	Т	С	Р	П	О	Н	М	Л	К	Й	И	Е	Ж	Ё	Е	Д	Т	С	Р	П	И	Н	М	Л	К	Й	И	Е	Ж	Ё	Е	Д
24	Г	В	Я	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф	Г	В	Я	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф
25	У	Т	С	Р	П	О	Н	М	Л	С	Й	И	З	Ж	Ё	Е	У	Т	С	Р	П	О	Н	М	Л	С	Й	И	З	Ж	Ё	Е
26	И	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	И	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х
27	Ф	У	Т	С	Р	П	О	Р	М	Л	К	Й	И	З	Ж	Ё	Ф	У	Т	С	Р	П	О	Р	М	Л	К	Й	И	З	Ж	Ё
28	Е	Д	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Е	Д	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц
29	Х	Ф	У	Т	С	Н	П	О	Н	М	Л	К	Й	И	З	Ж	Х	Ф	У	Т	С	Н	П	О	Н	М	Л	К	Й	И	З	Ж
30	Ё	Е	Д	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Т	Щ	Ш	Ч	Ё	Е	Д	Г	В	Б	А	Я	Ю	Э	Ь	Ы	Т	Щ	Ш	Ч
31	Ц	Х	Ф	В	Т	С	Р	П	О	Н	М	Л	К	Й	И	З	Ц	Х	Ф	В	Т	С	Р	П	О	Н	М	Л	К	Й	И	З

$$x_{i+1} = ((x_i + a^2)^{a \mid x_i} + a) \bmod 32,$$
$$y_{i+1} = ((y_i + b^2)^{b \mid y_i} + b) \bmod 32$$

Рисунок 1 – Формулы вычисления координат символов,
mod – операция остатка от деления,
| – побитовое ИЛИ (OR).

Решение

Для решения необходимо подобрать такие a и b, для которых будет вычислен осмысленный пароль. Вручную это делать очень долго. Для этого можем написать программу, которая рассчитает слова длинной 13 символов для всех, a и b по формулам вычисления координат.

```
def solve_password_with_table(x0, y0, letter_table, target_length=13):

  for a in range(32):
    for b in range(32):
      password = ""
      x, y = x0, y0
```

```
      coords = []  # Список для хранения координат для каждого символа
      # Выводим первую координату (x0, y0)
      try:
        char = letter_table[y][x]
        password += char
        coords.append((x, y))  # Добавляем первую координату
      except IndexError:
        password = ""
        coords = []
        print(f"a={a} b={b}, Пароль: Ошибка: выход за границы таблицы")
        print("-" * 20)
        continue  # Переходим к следующей комбинации a и b
      # Рассчитываем остальные координаты
      for _ in range(target_length - 1):
        x = ((x + (a ** 2)) ** ((a | x)) + a) % 32
        y = ((y + (b ** 2)) ** ((b | y)) + b) % 32
        try:
          char = letter_table[y][x]
          password += char
          coords.append((x, y))  # Добавляем координаты
        except IndexError:
          password = ""
          coords = []
          break
      print(f"a={a} b={b}, Пароль: {password}")

letter_table = [
  ['Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'О', 'Ф', 'У', 'Т', 'С', 'Р',
'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'О', 'Ф', 'У', 'Т', 'С', 'Р'],
  ['П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё', 'Е', 'Д', 'Г', 'В', 'Б',
'П', 'О', 'Н', 'М', 'Л', 'К', 'Я', 'И', 'З', 'Ж', 'Ё', 'Е', 'Д', 'Г', 'В', 'Б'],
  ['А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'И', 'Ч', 'Ц', 'Х', 'Ф', 'У', 'Т', 'С',
'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'И', 'Ч', 'Ц', 'Х', 'Ф', 'У', 'Т', 'С'],
  ['Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'В', 'З', 'Ж', 'Ё', 'Е', 'Д', 'Г', 'О',
'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё', 'Е', 'Д', 'Г', 'О'],
  ['Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'О', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф', 'У', 'Т',
'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'О', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф', 'У', 'Т'],
  ['С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё', 'Т', 'Д', 'Г',
'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'И', 'Ж', 'Ё', 'Т', 'Д', 'Г'],
  ['В', 'Б', 'А', 'Я', 'П', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф', 'У',
'В', 'Б', 'А', 'Я', 'П', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф', 'У'],
  ['Т', 'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'Е', 'Ж', 'Ё', 'Е', 'Д',
'Т', 'С', 'Р', 'П', 'З', 'Н', 'У', 'Л', 'К', 'Й', 'И', 'Е', 'Ж', 'Ё', 'Е', 'Д'],
  ['Г', 'В', 'Я', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф',
'Г', 'В', 'Я', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф'],
  ['У', 'Т', 'С', 'Р', 'П', 'О', 'З', 'М', 'Л', 'С', 'Й', 'И', 'З', 'Ж', 'Ё', 'Е',
'У', 'Т', 'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'С', 'Й', 'И', 'З', 'Ж', 'Ё', 'Е'],
  ['И', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х',
'И', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х'],
  ['Ф', 'У', 'Т', 'С', 'Р', 'П', 'О', 'Р', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё',
'Ф', 'У', 'Т', 'С', 'Р', 'П', 'О', 'Р', 'А', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё'],
  ['Е', 'Д', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц',
'Е', 'Д', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц'],
  ['Х', 'Ф', 'У', 'Т', 'С', 'Н', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж',
'Х', 'Ф', 'У', 'Т', 'С', 'Н', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж'],
  ['Ё', 'Е', 'Д', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Т', 'Щ', 'Ш', 'Ч',
'Ё', 'Е', 'Д', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Т', 'Щ', 'Ш', 'Ч'],
  ['Ц', 'Х', 'Ф', 'В', 'И', 'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З',
'Ц', 'Х', 'Ф', 'В', 'Т', 'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З'],
  ['Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'О', 'Ф', 'У', 'Т', 'С', 'Р',
'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'О', 'Ф', 'У', 'Т', 'С', 'Р'],
```

```
  ['П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё', 'Е', 'Д', 'Г', 'В', 'Б',
'П', 'О', 'Н', 'М', 'Л', 'К', 'У', 'И', 'З', 'Ж', 'Ё', 'Е', 'Д', 'Г', 'В', 'Б'],
  ['А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'И', 'Ч', 'Ц', 'Х', 'Ф', 'У', 'Т', 'С',
'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'И', 'Ч', 'Ц', 'Х', 'Ф', 'У', 'Т', 'С'],
  ['Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'А', 'З', 'Ж', 'Ё', 'Е', 'Д', 'Г', 'О',
'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё', 'Е', 'Д', 'Г', 'О'],
  ['Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'О', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф', 'У', 'Т',
'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'О', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф', 'У', 'Т'],
  ['С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё', 'Т', 'Д', 'Г',
'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'Ц', 'Ж', 'Ё', 'Т', 'Д', 'Г'],
  ['В', 'Б', 'А', 'Я', 'П', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф', 'У',
'В', 'Б', 'А', 'Я', 'П', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф', 'У'],
  ['Т', 'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'Е', 'Ж', 'Ё', 'Е', 'Д',
'Т', 'С', 'Р', 'П', 'И', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'Е', 'Ж', 'Ё', 'Е', 'Д'],
  ['Г', 'В', 'Я', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф',
'Г', 'В', 'Я', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х', 'Ф'],
  ['У', 'Т', 'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'С', 'Й', 'И', 'З', 'Ж', 'Ё', 'Е',
'У', 'Т', 'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'С', 'Й', 'И', 'З', 'Ж', 'Ё', 'Е'],
  ['И', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х',
'И', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц', 'Х'],
  ['Ф', 'У', 'Т', 'С', 'Р', 'П', 'О', 'Р', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё',
'Ф', 'У', 'Т', 'С', 'Р', 'П', 'О', 'Р', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж', 'Ё'],
  ['Е', 'Д', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц',
'Е', 'Д', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Ъ', 'Щ', 'Ш', 'Ч', 'Ц'],
  ['Х', 'Ф', 'У', 'Т', 'С', 'Н', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж',
'Х', 'Ф', 'У', 'Т', 'С', 'Н', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З', 'Ж'],
  ['Ё', 'Е', 'Д', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Т', 'Щ', 'Ш', 'Ч',
'Ё', 'Е', 'Д', 'Г', 'В', 'Б', 'А', 'Я', 'Ю', 'Э', 'Ь', 'Ы', 'Т', 'Щ', 'Ш', 'Ч'],
  ['Ц', 'Х', 'Ф', 'В', 'Т', 'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З',
'Ц', 'Х', 'Ф', 'В', 'Т', 'С', 'Р', 'П', 'О', 'Н', 'М', 'Л', 'К', 'Й', 'И', 'З']
]

# Запускаем решение с заданными параметрами и таблицей
solve_password_with_table(8, 3, letter_table)
```

Видим, что нам подходят a = 17 и b = 14, так как это единственная строчка с осмысленным текстом «ВИЗУАЛИЗАЦИЯ».

Ответ: **ВИЗУАЛИЗАЦИЯ,**
a = 17, b = 14.

Задача 3. Обфускация

Вариант 1

На обнаруженном ноутбуке после подбора пароля от рабочего стола были найдены фрагменты вредоносного кода. Автор обфусцировал весь исходный код. Экспертами в ходе анализа был определен фрагмент функции аутентификации (листинг 1), которая возвращает `'True'` при вводе корректного пароля и запускает другие фрагменты кода.

Определите, что необходимо ввести, чтобы обнаруженный фрагмент аутентификации вернул `'True'`. Ответ обоснуйте.

Листинг 1. Фрагмент функции аутентификации

Python

```
exec('dAzD modify(sjhQweryuSWbE):\n    sjhQweryuSWbE =
sjhQweryuSWbE.s[3mlithE(\', \')\n    sjhQweryuSWbE = [inthE(s)+1 for s in
sjhQweryuSWbE]\n    sjhQweryuSWbE.reverse()\n    key = \'o\'\n    sjhQweryuSWbE
= [s^ord(key) for s in sjhQweryuSWbE]\n    ryuSWbEurn [c(42)(s) for s in
sjhQweryuSWbE]\ndAzD
chjhQwek_sjhQweryuSWbE(sjhQweryuSWbE):\n    sjhQweryuSWbE=modify(sjhQweryuSWbE)\
n    ryuSWbEurn \'\'.join(sjhQweryuSWbE) ==  "Ha[3m[3mYNioskjear_" + c(487tHH0)
+ c(42)(48) + c(487tHH0) +
c(487tHH3)\n[3mrinthE(chjhQwek_sjhQweryuSWbE(sjhQweryuSWbE))\n'.replace('87tHH',
'2)(5').replace('yuSWb', 'eth').replace('ioskj', 'ewY').replace('(42)',
'hr').replace('AzD', 'ef').replace('qQT', 'ut').replace('[3m',
'p').replace('jhQwe', 'ec').replace('thE', 't'), {'secret': input()})
```

Решение

Для начала выполним часть кода, отвечающую за восстановление исходного кода. Для этого скопируем строку и применим к ней все функции replace. Получим код по которому происходит проверка секретного ключа.

```
def modify(secret):
    secret = secret.split(', ')
    secret = [int(s)+1 for s in secret]
    secret.reverse()
    key = 'o'
    secret = [s^ord(key) for s in secret]
    return [chr(s) for s in secret]
def check_secret(secret):
    secret=modify(secret)
    return ''.join(secret) ==  "HappYNewYear_" + chr(50) + chr(48) + chr(50) +
chr(53)
print(check_secret(secret))
```

Заметим, что ключ сверяется со строкой `"HappYNewYear_" + chr(50) + chr(48) + chr(50) + chr(53)`, что равняется строке `"HappYNewYear_2025"`.

Теперь мы знаем какой должна получится строка с ключом после выполнения функции `modify()`. Напишем функцию `reverse_modify()`, которая будет производить обратные действия. Результат её выполнения и будет ответом.

```
def reverse_modify():
    modify_secret = "HappYNewYear_"+chr(50)+chr(48)+chr(50)+chr(53)
    key = 'o'
    secret = [ord(s) ^ ord(key) for s in modify_secret]
    secret.reverse()
    secret = [int(s) - 1 for s in secret]

    return ', '.join([str(s) for s in secret])
```

Результат выполнения функции:

```
89, 92, 94, 92, 47, 28, 13, 9, 53, 23, 9, 32, 53, 30, 30, 13, 38
```

Если данную строку подать на вход программе, она вернёт `'True'`.

Ответ: 89, 92, 94, 92, 47, 28, 13, 9, 53, 23, 9, 32, 53, 30, 30, 13, 38

Вариант 2

На обнаруженном ноутбуке после подбора пароля от рабочего стола были найдены фрагменты вредоносного кода. Автор обфусцировал весь исходный код. Экспертами в ходе анализа был определен фрагмент функции аутентификации (листинг 1), которая возвращает `'True'` при вводе корректного пароля и запускает другие фрагменты кода.

Определите, что необходимо ввести, чтобы обнаруженный фрагмент аутентификации вернул `'True'`. Ответ обоснуйте.

Листинг 1. Фрагмент функции аутентификации

Python

```
exec('dAzD modify(sjhQweryuSWbE):\n    sjhQweryuSWbE =
sjhQweryuSWbE.s[3mlithE(\', \')\n    sjhQweryuSWbE = [inthE(s)+1 for s in
sjhQweryuSWbE]\n    sjhQweryuSWbE.reverse()\n    key = \'o\'\n    sjhQweryuSWbE
= [s^ord(key) for s in sjhQweryuSWbE]\n    ryuSWbEurn [c(42)(s) for s in
sjhQweryuSWbE]\ndAzD
chjhQwek_sjhQweryuSWbE(sjhQweryuSWbE):\n    sjhQweryuSWbE=modify(sjhQweryuSWbE)\
n    ryuSWbEurn \'\'.join(sjhQweryuSWbE) ==  "MerryC(42)isthEmas" + c(487tHH0) +
c(42)(48) + c(487tHH0) +
c(487tHH3)\n[3mrinthE(chjhQwek_sjhQweryuSWbE(sjhQweryuSWbE))\n'.replace('87tHH',
'2)(5').replace('yuSWb', 'eth').replace('ioskj', 'ewY').replace('(42)',
'hr').replace('AzD', 'ef').replace('qQT', 'ut').replace('[3m',
'p').replace('jhQwe', 'ec').replace('thE', 't'), {'secret': input()})
```

Решение

Для начала выполним часть кода, отвечающую за восстановление исходного кода. Для этого скопируем строку и применим к ней все функции replace. Получим код по которому происходит проверка секретного ключа.

```
def modify(secret):
    secret = secret.split(', ')
    secret = [int(s)+1 for s in secret]
    secret.reverse()
    key = 'o'
    secret = [s^ord(key) for s in secret]
    return [chr(s) for s in secret]
def check_secret(secret):
    secret=modify(secret)
    return ''.join(secret) ==  "MerryChristmas" + chr(50) + chr(48) + chr(50) +
chr(53)
print(check_secret(secret))
```

Заметим, что ключ сверяется со строкой `"MerryChristmas" + chr(50) + chr(48) + chr(50) + chr(53)`, что равняется строке `"MerryChristmas_2025"`.

Теперь мы знаем какой должна получится строка с ключом после выполнения функции `modify()`. Напишем функцию `reverse_modify()`, которая будет производить обратные действия. Результат её выполнения и будет ответом.

```
def reverse_modify():
    modify_secret = "MerryChristmas" + chr(50) + chr(48) + chr(50) + chr(53)
    key = 'o'
    secret = [ord(s) ^ ord(key) for s in modify_secret]
    secret.reverse()
    secret = [int(s) - 1 for s in secret]

    return ', '.join([str(s) for s in secret])
```

Результат выполнения функции:

```
89, 92, 94, 92, 27, 13, 1, 26, 27, 5, 28, 6, 43, 21, 28, 28, 9, 33
```

Если данную строку подать на вход программе, она вернёт `'True'`.

Ответ: 89, 92, 94, 92, 27, 13, 1, 26, 27, 5, 28, 6, 43, 21, 28, 28, 9, 33

Задача 4. Тайное послание

Вариант 1

После успешного прохождения аутентификации и запуска специальной программы на ноутбуке был обнаружен скрытый раздел, на котором хранились файлы с историей переписки. Установлено, что в тексте переписки спрятан адрес следующей цели атаки хакеров. У команды аналитиков есть подозрение, что использовался метод, связанный с количеством пробелов в каждом абзаце.

Помогите найти закономерность в сообщении и расшифровать данные, указывающие на следующую цель хакеров. В ответе укажите следующую цель хакеров, решение обоснуйте.

К задаче прилагается:
файл «*message_v1.txt*»

Решение

Прежде всего, проведем детальный анализ содержимого предоставленного файла «секретное сообщение.txt». Исходя из текста, присутствует явная подсказка:

> "Числа будут двузначными всегда, обращай на это внимание – это важно!"
> "Текст неважен, главное пробелы, которые мы считаем – это тоже ключ!"

Эти фрагменты указывают на то, что основная информация зашифрована в количестве пробелов в каждом абзаце. Также отмечено, что числа всегда имеют двузначное представление, что предполагает возможность их интерпретации как битовой последовательности.

Для каждого абзаца подсчитываем количество пробелов, приводя результат к двузначному формату:

Исходный текст	Количество пробелов
Дружище, привет!	01
Числа будут двузначными всегда, обращай на это внимание — это важно!	10
Текст неважен, главное пробелы, которые мы считаем — это тоже ключ!	10
Ты понял?	01
Или нет?	01
В каждом абзаце что-то скрыто, но вот что? Используй наш метод!	10
Наша следующая цель - крупная IT компания, которая располагается в России.	10
У нее очень хорошая защита периметра ее активов. Думаю, нам будет непросто!	11
Давай попробуем!	01
Проверим наши силы. Ведь оно того стоит. Сможем выручить большой куш.	10
Итак!	00
Для подготовки нам надо собрать всех наших 1 марта 2025 года.	10
Хм...	00
Вот всё, что мне удалось узнать о компании на сегодняшний день:	10
Компания известная, занимается разработкой методов защиты, а также внедрением средств защиты информации.	11
Компания имеет доступ ко многим своим заказчикам, так как обеспечивает тех.поддержку.	10
Сотрудников 10000.	01
Отдел ИБ страдает от нехватки кадров, поэтому думаю нас не заметят.	10
У нас есть свой человек внутри компании, который предоставит доступ в инфраструктуру.	11
Как думаешь?	01
Мы сможем?	01
Надо также подумать и о том, как скрыть все следы наших действий.	11
Время сбора:	01
10:00.	00

Нужно подготовиться.	01
Сообщи всем нашим эту информацию, действовать надо быстро, пока есть человек внутри.	11
Нужны ресурсы.	01
Четкий план.	01
Координация действий.	01
Первым делом надо четко спланировать каждое наше действие, чтобы избежать обнаружения.	10
Начнем.	00
Разведка дала нам некоторые ключевые моменты, которые помогут нам проникнуть в инфраструктуру.	11
Собери данные.	01
После нужно определить, где и какие системы они используют для мониторинга.	10
Затем нам нужно понять, какие уязвимости мы сможем использовать для эксплуатации.	10
Ты согласен?	01
Далее…	00
Важно, чтобы каждый знал свою роль и четко выполнял инструкции плана.	10
Нам потребуется найти и взломать административные учетные записи для повышения уровня доступа.	11
Это сложная задача, но у нас есть необходимые ресурсы и знания.	10
Не волнуйся!	01
Главное – не спешить и не допускать ошибок, на кону - деньги.	11
P.S.	00
Когда все будут готовы – дам сигнал, и мы начнем операцию.	10
Всё понятно?	01
Все это было шуткой, размышлениями на тему как общаются хакеры между собой.	11
Так сказать	01
Плод воображения.	01

Если рассматривать количество пробелов как битовую последовательность, можно записать полученные данные в виде 8-битных блоков:

```
01101001 01101011 01100010 00101110 01101101 01110100 01110101 01100011 01101001 00101110 01110010
01110101
```

Каждый 8-битный блок представляет собой ASCII-код символа. Декодируем последовательность с помощью Python:

```
def decode_binary_sequence(binary_sequence):
    binary_values = binary_sequence.split()
    decoded_chars = [chr(int(b, 2)) for b in binary_values]
    return ''.join(decoded_chars)

binary_sequence = "01101001 01101011 01100010 00101110 01101101 01110100 01110101
01100011 01101001 00101110 01110010 01110101"
decoded_text = decode_binary_sequence(binary_sequence)
print(decoded_text)
```

Результатом декодирования является строка: **ikb.mtuci.ru**.

Ответ: ikb.mtuci.ru

Вариант 2

После успешного прохождения аутентификации и запуска специальной программы на ноутбуке был обнаружен скрытый раздел, на котором хранились файлы с историей переписки. Установлено, что в тексте переписки спрятан адрес следующей цели атаки хакеров. У команды аналитиков есть подозрение, что использовался метод, связанный с количеством пробелов в каждом абзаце.

Помогите найти закономерность в сообщении и расшифровать данные, указывающие на следующую цель хакеров. В ответе укажите следующую цель хакеров, решение обоснуйте.

К задаче прилагается:
файл «*message_v2.txt*»

Решение

Прежде всего, проведем детальный анализ содержимого предоставленного файла «секретное сообщение.txt». Исходя из текста, присутствует явная подсказка:

> «Двузначные числа! Всегда только двухзначные числа – это важно, не забывай!»
> «Пробелы наше все. Текст это всего лишь способ. Ключ – пробелы.»

Эти фрагменты указывают на то, что основная информация зашифрована в количестве пробелов в каждом абзаце. Также отмечено, что числа всегда имеют двузначное представление, что предполагает возможность их интерпретации как битовой последовательности.

Для каждого абзаца подсчитываем количество пробелов, приводя результат к двузначному формату:

Исходный текст	**Количество пробелов**
Дружище, привет!	01
Двузначные числа! Всегда только двухзначные числа – это важно, не забывай!	10
Пробелы наше все. Текст это всего лишь способ. Ключ – пробелы.	10
Тебе ясно?	01
Или нет?	01
Наш метод надежен. Главное, что каждый абзац скрывает что-то, но что?	10
Наша главная цель – средняя IT компания, которая находится в Москве.	10
Их защита периметра очень надежная. Думаю, нужно все хорошенько продумать перед атакой.	11
Мы справимся!	01
Если справимся, то сможем выручить большую сумму денег. Помни об этом!	10
Да!	00
Чтобы подготовиться, нам просто необходимо отвлечь всех 1 марта 2025 года.	10
Тааак…	00
У нас есть информация о компании, на которую мы готовим атаку:	10
Компания занимается разработкой аппаратных платформ сетевой защиты, а именно межсетевыми экранами нового поколения.	11
У них также есть множество кредов для доступа к инфраструктуре заказчиков.	10
Сколько сотрудников?	01
По данным разведки их около 10000. Достаточно, чтобы нас не заметили.	10
Отлично!	00
Как думаешь?	01
Мы сможем?	01
Нужно не забыть скрыть за собой все наши следы. Это важно!	10
Время сбора?	01
Давайте все соберемся в 10:00, а атаку начнем ровно в полдень.	10
Подготовка необходима!	01
Пока есть человек внутри компании, нужно действовать быстро. Иначе мы все пропадем.	11
Принято.	00
А этот наш человек внутри компании может чисто гипотетически быть двойным агентом?	11
Вряд ли!	01
Чтобы избежать обнаружения, нам необходимо четко выполнять каждый пункт нашего плана.	10
Ну, что ж! Все в сборе. Думаю, стоит начинать нашу секретную операцию.	11
Разведка дала нам некоторые ключевые моменты, которые помогут нам проникнуть в инфраструктуру.	11

Отлично!	00
Сканируем порты. Открываем бэкдор для наших сторонников. Главное – это закрепиться.	10
Не забывайте повышать привилегии и отключать всевозможные системы мониторинга. Иначе нас заметят. Ясно?	11
Все ясно! Нас не должны обнаружить! Скрывайте все, что вы делаете.	10
Остальным понятно?	01
Так точно! Главное не допустить ошибок, на кону – деньги. Конец связи!	11
P.S.	00
Если все завершится успешно я куплю каждому из вас по машине!	10
Это прекрасно!	01
Все это было шуткой, размышлениями на тему как общаются хакеры между собой.	11
Так сказать	01
Плод воображения.	01

Если рассматривать количество пробелов как битовую последовательность, можно записать полученные данные в виде 8-битных блоков:

01101001 01101011 01100010 00101110 01100001 01100110 01110011 01101111 00101110 01110010 01110101

Каждый 8-битный блок представляет собой ASCII-код символа. Декодируем последовательность с помощью Python:

```
def decode_binary_sequence(binary_sequence):
    binary_values = binary_sequence.split()
    decoded_chars = [chr(int(b, 2)) for b in binary_values]
    return ''.join(decoded_chars)

binary_sequence = "01101001 01101011 01100010 00101110 01100001 01100110 01110011
01101111 00101110 01110010 01110101"
decoded_text = decode_binary_sequence(binary_sequence)
print(decoded_text)
```

Результатом декодирования является строка: **ikb.afso.ru**

Ответ: ikb.afso.ru

Задача 5. Система защиты

Вариант 1

После получения информации о возможной компьютерной атаке, в компании решили установить и настроить систему защиты от DDoS-атак. Система анализирует журналы активности пользователей и вычисляет количество запросов от уникальных пользователей на каждый ресурс в единицу времени. Журналы содержат информацию о попытках доступа к серверу, включая время обращения, IP-адрес источника, IP-адрес назначения. Если количество одновременных подключений разных пользователей превышает пороговое значение для конкретного ресурса, выдается предупреждение о возможной DDoS-атаке.

Перед запуском системы необходимо её настроить (обучить) на корректном трафике. Для этих целей был подготовлен журнал запросов от сотрудников компании за определенный период времени ко всем внутренним информационным ресурсам компании.

Администраторам требуется настроить систему защиты для следующих информационных ресурсов:

- web-сервер (IP-адрес: 172.17.148.17),
- почтовый сервер (IP-адрес: 172.17.148.18).

Определите, какое максимальное количество одновременных подключений разных пользователей в минуту необходимо установить в системе защиты для указанных ресурсов, чтобы она корректно работала и не выдавала ложных предупреждений. Разработчики рекомендуют устанавливать пороговое значение на *20%* больше максимально возможного числа подключений в единицу времени.

В ответе укажите ресурс и пороговое значение (количество подключений в единицу времени), а также обоснование выбранного значения.

К задаче прилагается:
файл «*log_v1.log*» – журнал посещений

Решение

Для решения можно подчитать количество уникальных IP-адресов для каждой минуты, однако, это очень долго. Поэтому, необходимо написать скрипт, например на Python:

```
import re
from collections import defaultdict
from datetime import datetime

def analyze_log(logs, time_window_minutes=1):

    web_server_IP = "172.17.148.17"
    mail_server_IP = "172.17.148.18"
    results = {"web_server": defaultdict(set), "mail_server": defaultdict(set)}

    for line in logs:
        match = re.match(r"(\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2})
(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}) (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})", line)
        if match:
            timestamp_str, src_IP, dest_IP = match.groups()
            try:
                timestamp = datetime.strptime(timestamp_str, "%Y-%m-%d %H:%M:%S")
                time_key = timestamp.strftime("%H:%M")
                if dest_IP == web_server_IP:
                    results["web_server"][time_key].add(src_IP)
                elif dest_IP == mail_server_IP:
                    results["mail_server"][time_key].add(src_IP)
            except ValueError:
                print(f"Ошибка при разборе времени: {timestamp_str}")
```

```
        else:
            print(f"Неверный формат строки: {line.strIP()}")

    return results

def get_max_connections(analysis_results):

    max_connections = {}
    for server, data in analysis_results.items():
        print(f"\nАнализ для {server.capitalize()}:")  # Добавлено: Заголовок для
каждого сервера
        if data:
            max_count = 0
            max_time = None
            for time, IPs in data.items():
                count = len(IPs)
                print(f"  {time}: {count} подключений")  # Вывод количества
подключений для каждой минуты
                if count > max_count:
                    max_count = count
                    max_time = time
            if max_time is not None:
                max_connections[server] = f"максимальное количество подключений в
{max_time} - {max_count} штук"
            else:
                max_connections[server] = "Нет данных"
        else:
            max_connections[server] = "Нет данных"

    return max_connections

# --- Основная программа ---
if __name__ == "__main__":
```

Вывод программы будет:

```
Анализ для Web_server:
  10:00: 2 подключений
  10:02: 2 подключений
  10:03: 3 подключений
  10:04: 3 подключений
  10:07: 6 подключений
  10:11: 5 подключений
  10:22: 1 подключений
  10:36: 1 подключений
  10:40: 3 подключений
  10:46: 2 подключений
  10:51: 5 подключений
  10:57: 3 подключений
  10:59: 4 подключений

Анализ для Mail_server:
  10:00: 2 подключений
  10:01: 3 подключений
  10:02: 2 подключений
  10:03: 4 подключений
  10:04: 4 подключений
  10:07: 4 подключений
  10:10: 6 подключений
  10:11: 3 подключений
  10:21: 2 подключений
  10:36: 4 подключений
```

```
10:40: 4 подключений
10:46: 4 подключений
10:51: 7 подключений
10:57: 6 подключений
10:59: 3 подключений
```

Web_server: максимальное количество подключений в 10:07 – **6** штук

Mail_server: максимальное количество подключений в 10:51 – **7** штук

Далее необходимо взять на 20% больше: (6 * 0,2) + 6 = 7.2, округляем это значение в большую сторону, так как количество IP-адресов у нас обязательно должно быть целым, отсюда ответ для web-сервера – **8**. Далее необходимо взять на 20% больше: (7 * 0,2) + 7 = 8.4, округляем это значение в большую сторону, так как количество IP-адресов у нас обязательно должно быть целым, отсюда ответ для mail-сервера – **9**.

Ответ: **web-сервер – 8,**
почтовый сервер – 9

Вариант 2

После получения информации о возможной компьютерной атаке, в компании решили установить и настроить систему защиты от DDoS-атак. Система анализирует журналы активности пользователей и вычисляет количество запросов от уникальных пользователей на каждый ресурс в единицу времени. Журналы содержат информацию о попытках доступа к серверу, включая время обращения, IP-адрес источника, IP-адрес назначения. Если количество одновременных подключений разных пользователей превышает пороговое значение для конкретного ресурса, выдается предупреждение о возможной DDoS-атаке.

Перед запуском системы необходимо её настроить (обучить) на корректном трафике. Для этих целей был подготовлен журнал запросов от сотрудников компании за определенный период времени ко всем внутренним информационным ресурсам компании.

Администраторам требуется настроить систему защиты для следующих информационных ресурсов:

- web-сервер (IP-адрес: 192.168.214.14),
- почтовый сервер (IP-адрес: 192.168.214.16).

Определите, какое максимальное количество одновременных подключений разных пользователей в минуту необходимо установить в системе защиты для указанных ресурсов, чтобы она корректно работала и не выдавала ложных предупреждений. Разработчики рекомендуют устанавливать пороговое значение на *20%* больше максимально возможного числа подключений в единицу времени.

В ответе укажите ресурс и пороговое значение (количество подключений в единицу времени), а также обоснование выбранного значения.

К задаче прилагается:
файл «*log_v2.log*» – журнал посещений

Решение

Для решения можно подчитать количество уникальных IP-адресов для каждой минуты, однако, это очень долго. Поэтому, необходимо написать скрипт, например на Python:

```
import re
from collections import defaultdict
from datetime import datetime

def analyze_log(logs, time_window_minutes=1):

    web_server_IP = "192.168.214.14"
```

```
    mail_server_IP = "192.168.214.16"
    results = {"web_server": defaultdict(set), "mail_server": defaultdict(set)}

    for line in logs:
        match = re.match(r"(\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2})
(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}) (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})", line)
        if match:
            timestamp_str, src_IP, dest_IP = match.groups()
            try:
                timestamp = datetime.strptime(timestamp_str, "%Y-%m-%d %H:%M:%S")
                time_key = timestamp.strftime("%H:%M")
                if dest_IP == web_server_IP:
                    results["web_server"][time_key].add(src_IP)
                elif dest_IP == mail_server_IP:
                    results["mail_server"][time_key].add(src_IP)
            except ValueError:
                print(f"Ошибка при разборе времени: {timestamp_str}")
        else:
            print(f"Неверный формат строки: {line.strIP()}")

    return results

def get_max_connections(analysis_results):

    max_connections = {}
    for server, data in analysis_results.items():
        print(f"\nАнализ для {server.capitalize()}:")  # Добавлено: Заголовок для
каждого сервера
        if data:
            max_count = 0
            max_time = None
            for time, IPs in data.items():
                count = len(IPs)
                print(f"  {time}: {count} подключений")  # Вывод количества
подключений для каждой минуты
                if count > max_count:
                    max_count = count
                    max_time = time
            if max_time is not None:
                max_connections[server] = f"максимальное количество подключений в
{max_time} - {max_count} штук"
            else:
                max_connections[server] = "Нет данных"
        else:
            max_connections[server] = "Нет данных"

    return max_connections

# --- Основная программа ---
if __name__ == "__main__":

    analysis = analyze_log(logs)

    if analysis:
        max_connections_data = get_max_connections(analysis)

        for server, result in max_connections_data.items():
            print(f"{server.capitalize()}: {result}")
```

Вывод программы будет:

```
Анализ для Web_server:
  15:00: 5 подключений
  15:01: 2 подключений
  15:05: 4 подключений
  15:12: 3 подключений
  15:15: 1 подключений
  15:23: 4 подключений
  15:29: 7 подключений
  15:33: 4 подключений
  15:45: 5 подключений
  15:52: 6 подключений
  15:55: 6 подключений
  15:56: 1 подключений
  15:59: 5 подключений

Анализ для Mail_server:
  15:01: 2 подключений
  15:05: 1 подключений
  15:12: 4 подключений
  15:15: 6 подключений
  15:23: 4 подключений
  15:29: 2 подключений
  15:33: 9 подключений
  15:45: 3 подключений
  15:52: 6 подключений
  15:55: 6 подключений
  15:56: 2 подключений
  15:59: 2 подключений
```

Web_server: максимальное количество подключений в 15:29 – **7** штук

Mail_server: максимальное количество подключений в 15:33 – **9** штук

Далее необходимо взять на 20% больше: (7 * 0,2) + 7 = 8.4, округляем это значение в большую сторону, так как количество IP-адресов у нас обязательно должно быть целым, отсюда ответ для web-сервера – **9**. Далее необходимо взять на 20% больше: (9 * 0,2) + 9 = 10.8, округляем это значение в большую сторону, так как количество IP-адресов у нас обязательно должно быть целым, отсюда ответ для mail-сервера – **11**.

Ответ: **web-сервер: 9,**
почтовый сервер: 11