Поймай шпиона, если сможешь, или Космический Котик III

Задание 1

Сначала необходимо определить, в каких именно пакетах содержится сообщение.

Вообще про стеганографические методы передачи информации было написано, что в случае ICMP пакетов (которые очень удобно использовать для передачи такой информации, потому что в сети обычно таких сообщений очень много, поэтому трудно порой понять, что с ними что-то не так) используются несколько подходов, в том числе:

- помещение данных в ICMP data для тех пакетов, которые являются failed-реквестами (в приложенном дампе такие есть, но данные в них не особо о чем-то говорят, так что этот подход не подойдет).
- помещение данных в ICMP data для тех пакетов, у которых необычный ID. Например, у нас это ID = 1984.

Итак, всего в сетевых логах у нас 32 пакета с таким ID, первый из которых - 881.

Позже (во втором задании) поясню, почему в 881 уже содержится часть сообщения. А пока, ответ на 1й вопрос - это 881 (впрочем, можно взять и любой другой из оставшихся 31 пакетов).

Ответ: 881

Задание 2

В ходе выполнения задания было несколько идей, как сообщение может шифроваться (тут напишу основные подходы, а подробно распишу только про правильный).

Вообще можно заметить, что у каждой пары *request* - *reply* поле data одно и то же, так что можно рассматривать только реквесты.

У каждого реквеста эти данные по длине составляют 662 символа. Причем большая часть из них совпадает.

Итак, какие были основные подходы:

- Рассматривать каждые 2 подряд идущих реквеста и символы, которые у них совпадают.
- Рассматривать каждые 2 подряд идущих реквеста и символы, которые в них различаются.
- Рассматривать несколько подряд идущих реквестов и смотреть на совпадающие символы.
- Рассматривать определенную алфавитную строку (алфавит от "a" до "w", повторенную много раз, длины тоже 662) и справнивать с ней данные каждого реквеста. Смотреть, какие символы различаются.

  Это был самый нормальный подход, потому что данные во всех пакетах почти на всех позициях совпадают с таким вот "закрученным" алфавитом.

  Кстати, этот подход как раз дал необходимое число пакетов для расшифровки.

  Ниже перечислю то, что выводилось в таком сравнении для каждого из пакетов:

  1. ceglonnhktwihjtkwbnqwdjncruihuthhtgfklm‘cbkonwjvcginqpbeqtgfisuwfhjpruwv‘gi
  2. ‘bejvllonnwehjqutihjvmowvbejmqpv‘fjcedfprughktwhsgvkjmlv‘lcejvcmgiqpubhqrtdfkpsdghpu‘gi

3. cbjlonwcihjoqbdijlpcbjmnv‘epv‘gfjscdfiugsqutdhkjpsvgihkjltvmv‘cbklonweqciqpsubqrfsudgihsru‘gfi
4. cjnmlonmw‘cihl‘cbdijlonpuwbgjsjlv‘cgfjqscdiobnsreikquwkjsgihkltchkotwvbkjlcbonqcbeqsbdgfqrfihqrutdihkjsrt‘gfi
5. bjrvmlonwcihm‘dkjnuwvbemlns‘cefjloqprw‘cbgfjgqsgnsuefhqsrtdikpkwgihl‘ckluchv‘jmlvcm‘clofhsrbgfiqsruedgfijqpsugfjr‘g
6. jnvlonwcfihkjmnwv‘edkjmquv‘edhlqptwdgfkqpsrw‘dfijrdgqprbinusufihutghihrcikjouvm‘mv‘mln‘celpghsbednqpedgjjv‘g
7. ‘crmloncfihjrutv‘ihlpucgksfrgiqsrubfhriufgsutikwbjnvjmovbenqhbhntkjwjtvl
8. bloncehqtdhtckowbjnvfgqsbefhqtdpgsrfisruikowncejonebsfihqpughswfjtvcl
9. ceglonnhktwihjtkwbnqwdjncruihuthhtgfklm‘cbkonwjvcginqpbeqtgfisuwfhjpruwv‘gi

Так вот, 9-е сообщение совпадает с 1-ым, точно так же дальше 10-2 – со 2-м и т.д.

Это показано, что подход уже более-менее правильный.

И нам нужно **восемь пакетов** (либо первые 8 реквестов взять, либо можно вторые восемь тоже).

Однако использование именно символов, к сожалению, ничего не дало, зато подошел следующий подход:

- Сравниваем данные из пакета с "закрученным алфавитом", как и в предыдущем пункте. Но не выписываем символы, а ставим в позицию 0, если данные совпадают с алфавитом, и 1, если данные не совпадают.

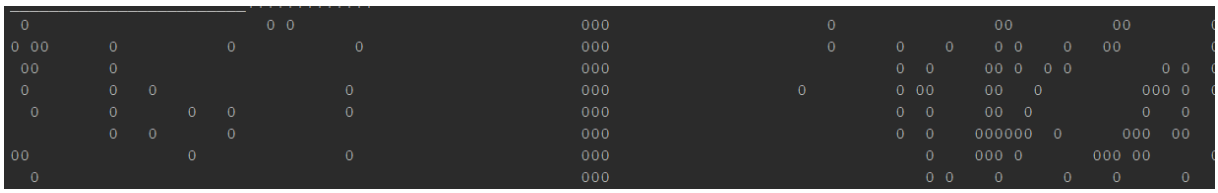
  Итого, получим строку длины 662, состоящую из 0 и 1. И таких нам нужно 8 строк (сформированных из данных в 8 идущих подряд пакетах).

  Дальнейшие действия описаны в задании 3.

Ответ: 8 пакетов.

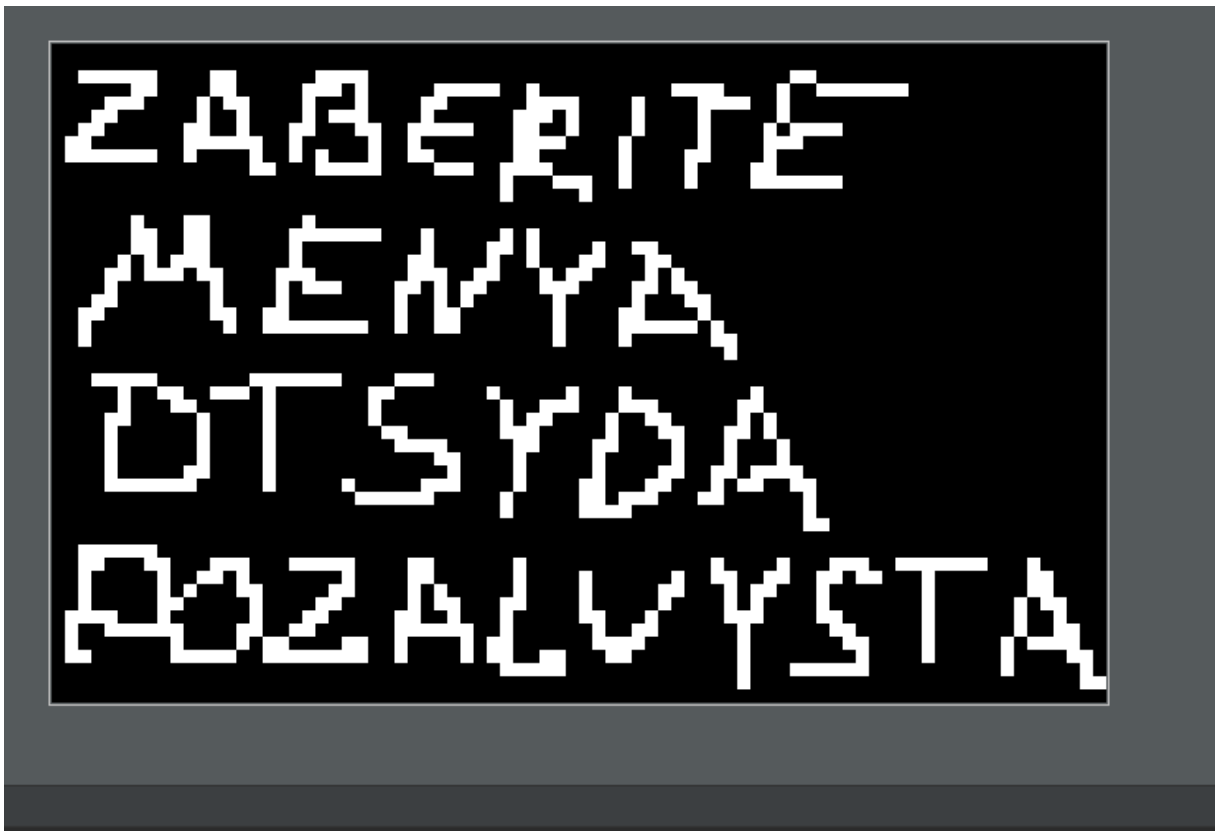
Задание 3

Итак, у нас есть 8 строк длины 662, состоящие из нулей и единиц. Вообще, если заменить нули на пробелы, а единицы на какие-нибудь стоящие символы, например нули, то появится ощущение, что зашифровано какое-то изображение (то есть в наших сообщениях хранится битовое изображение):



Вообще, так как строк (каждая строка соответствует пакету, как описывалось выше) получилось 8, то можно также предположить, что у нас есть 662 байта (то есть символы кажой из строк на определенной позиции - это биты одного байта: так, в каждой строке на нулевой позиции стоят биты нулевого байта).

Поэтому, дальше по коду (см. файл .java) я нахожу эти байты, а затем за счет встроенной в Java функции перевожу байты в изображение формата "BMP":



Ответ: ZABERITE MENYA OTSYDA POZALUYSTA