# **Домашнее задание к занятию 1.2 Сетевые модели TCP/IP и OSI**

## **Цель задания**

1. Изучить сетевую модель TCP\IP на практике.
2. Установить и поработать со сниффером сетевого трафика Wireshark.

### **Подготовка**

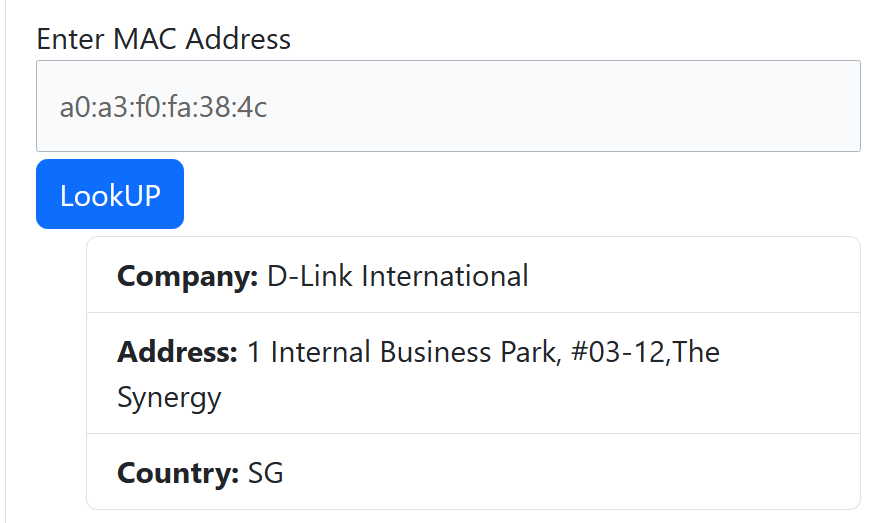
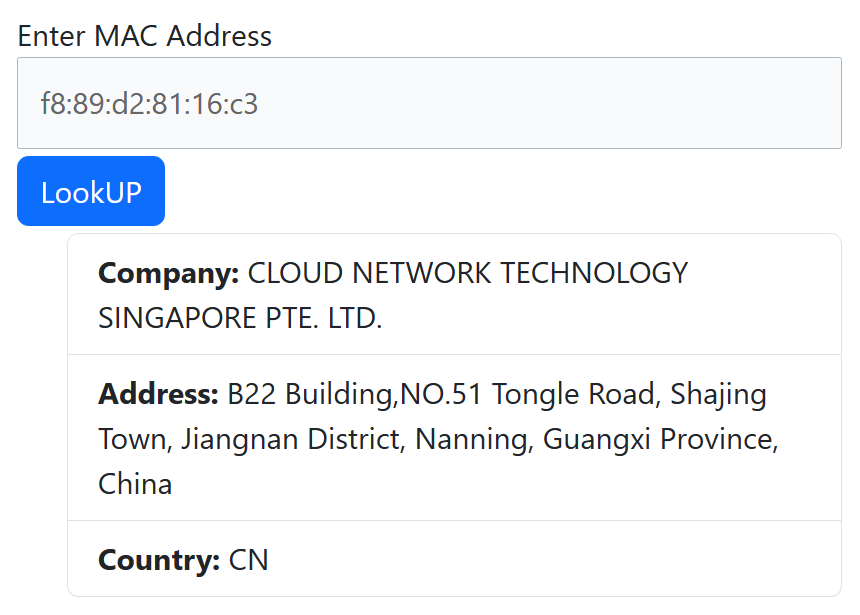
1. Установите программу Wireshark;
2. Запустите захват трафика с сетевого интерфейса;
3. Запустите браузер и зайдите на любой сайт.

## **Задание 1 (Ethernet).**

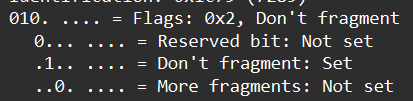
1. Выберите несколько пакетов из захваченного трафика;
2. Откройте уровень Ethernet;
3. Определите по MAC-адресу нескольких производителей сетевого оборудования.
4. 

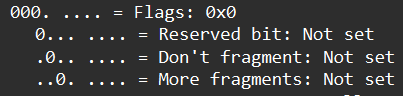


или



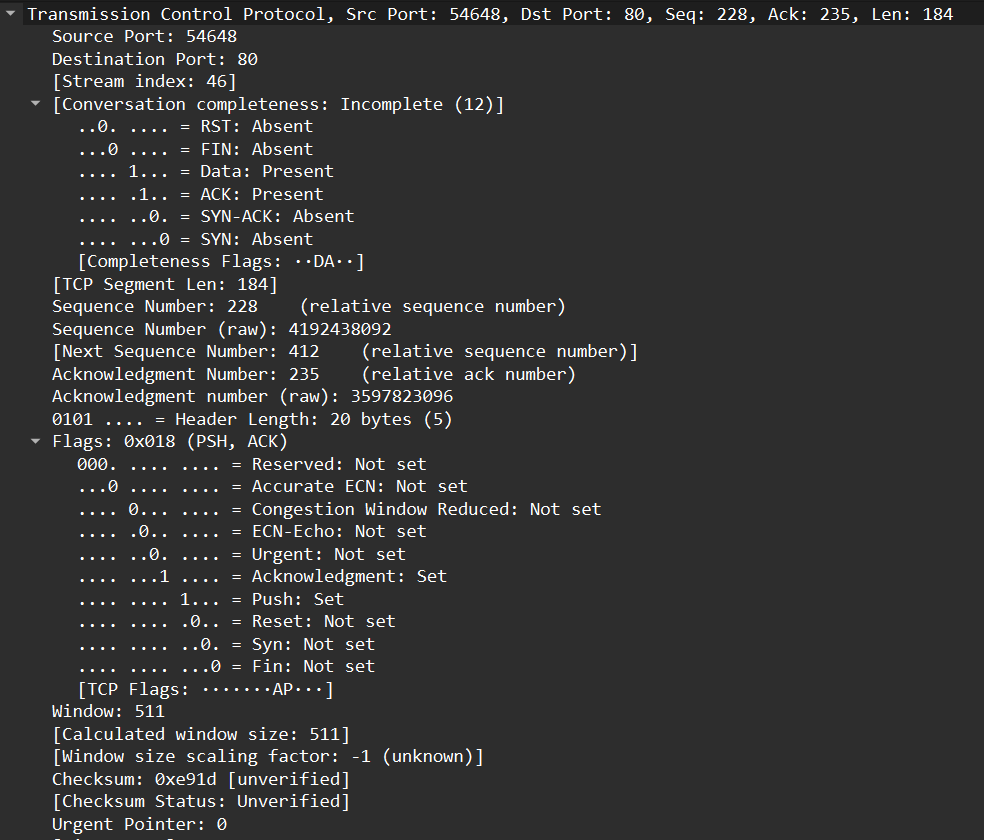
## **Задание 2 (IP).**

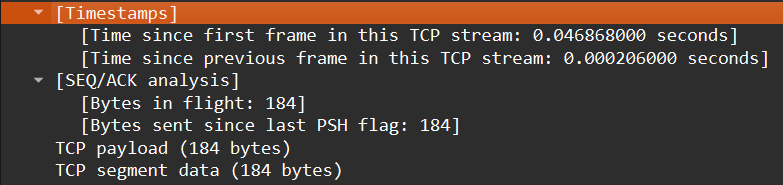
1. Выберите несколько пакетов из захваченного трафика;
2. Откройте уровень IP;
3. Какие флаги протокола IP вам удалось обнаружить?
4. 

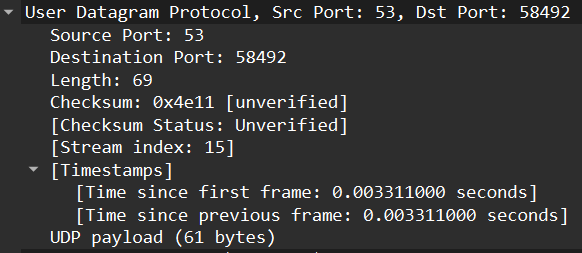


## **Задание 3 (TCP, UDP).**

1. Выберите несколько UDP (DNS или мессенджеры) и TCP (http) пакетов из захваченного трафика;
2. Откройте уровень TCP\UDP;
3. Сравните пакеты TCP и UDP между собой. В чём отличия?
4. Приведите ответ в свободной форме.







У TCP пакетов есть флаги, которые отражают состояния соединения, необходимые, как я правильно понимаю, для установки соединения и его “подтверждения”.

Также в TCP есть какие-то порядковые номера, которые, как было рассказано в интернетах, “позволяют модулю TCP получателя восстановить порядок полученных данных и удалить дубликаты пакетов.”

Посмотрев некоторое кол-во пакетов этих двух протоколов можно предположить, что из-за того, что в TCP протоколе больше данных, то полезная нагрузка в tcp и udp будет отличаться, если отправлять одинаковое кол-во данных изначально. Предположу, что и на “временные метки” (timestamps) бОльшее кол-во данных будет влиять.

## **Задание 4 (HTTP).**

1. Перейдите на сайт kremlin.ru;
2. Выберите любой http-поток (через меню, открывающееся по нажатию правой кнопки мыши, - следовать - поток HTTP);
3. Какой поток у вас открылся на самом деле (посмотрите в строке ввода фильтров)? Почему?
4. Открылся tcp.stream, предположу, что это потому что http инкапсулирован в tcp

“вышестоящие уровни пользуются услугами нижестоящих”

HTTP - прикладного уровня - 7

TCP - транспортного уровня - 4