# Сетевые и интернет технологии

## ДЗ №1: Примеры заголовков запроса/отклика (10шт на каждый тип)

\*aka «10&10 Headers of Request/Response»

Request:

1. **accept**: указывает тип информации, который должен прийти в качестве ответа от сервера;
2. **authority**: один из псевдо-заголовков протокола HTTP/2.x, вместе содержащих те же данные, что и *начальная строка запроса*, т.е. URI, метод и код состояния запроса. Данный заголовок аналогичен заголовку **host**, т.е. хранит домен, для которого предназначен запрос;
3. **cache-control** (общий заголовок): задаёт команды, обязательные для всех процессов кеширования;
4. **connection** (общий заголовок): указывает, останется ли данное соединение активным после завершения запроса (возможно ли будет выполнять последующие запросы на том же сервере);
5. **content-length**: указывает необходимую длину сообщения в байтах;
6. **expec**t: позволяет клиенту задать поведение сервера, передать сообщение, побуждающие к тем или иным действиям;
7. **if-none-match**: команды в поле выполняется клиентом, если указанные в поле значения не совпадают с содержимым поля etag, т.е. указанные объекты не совпадают с текущим;
8. **referer**: содержит URI ресурса, с которого клиент перешел на текущий ресурс;
9. **TE**: содержит список способов кодирования передаваемых данных, поддерживаемых клиентом;
10. **user-agent**: содержит полную информацию о клиенте (браузер, операционная система и т.д.).

Response:

1. **age**: указывает число секунд с последнего изменения ресурса;
2. **allow**: указывает методы, которые может использовать клиент;
3. **alternates**: указывает альтернативные способы представления ресурса (обычно не используется);
4. **content-disposition** (общий заголовок): индикатор того, что ответ будет отображаться в виде веб-страницы или её части;
5. **content-MD5**: используется для проверки целостности объектов сообщений;
6. **content-security-policy**: указывает, какие данные пользователь может загружать на данный ресурс;
7. **derived-from**: аналог content-version (указывает текущую версию HTTP-объекта);
8. **etag**: указывает версию кеша HTTP-объекта. Другими словами, хранит идентификатор загружаемого интернет-ресурса
9. **link** (общий заголовок): указывает на логически связанный с объектов ресурс (редко используется);
10. **vary**: индикатор того, что HTTP-объекта имеет несколько источников, и его содержимое в данный момент зависит от объекта, указанного в URI;

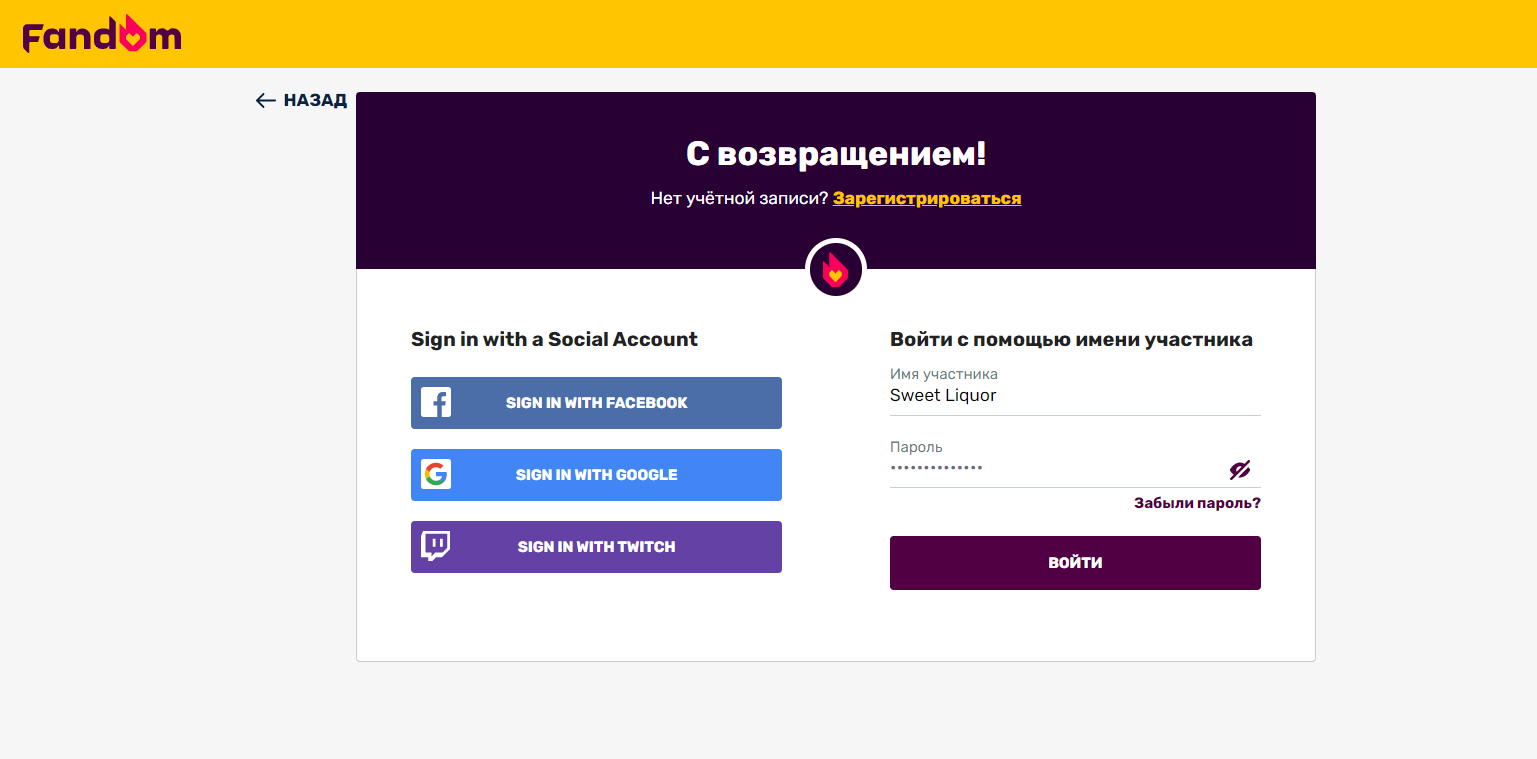
## ДЗ №2: Отслеживание данных при авторизации

\*aka «2 Socials Authentication»

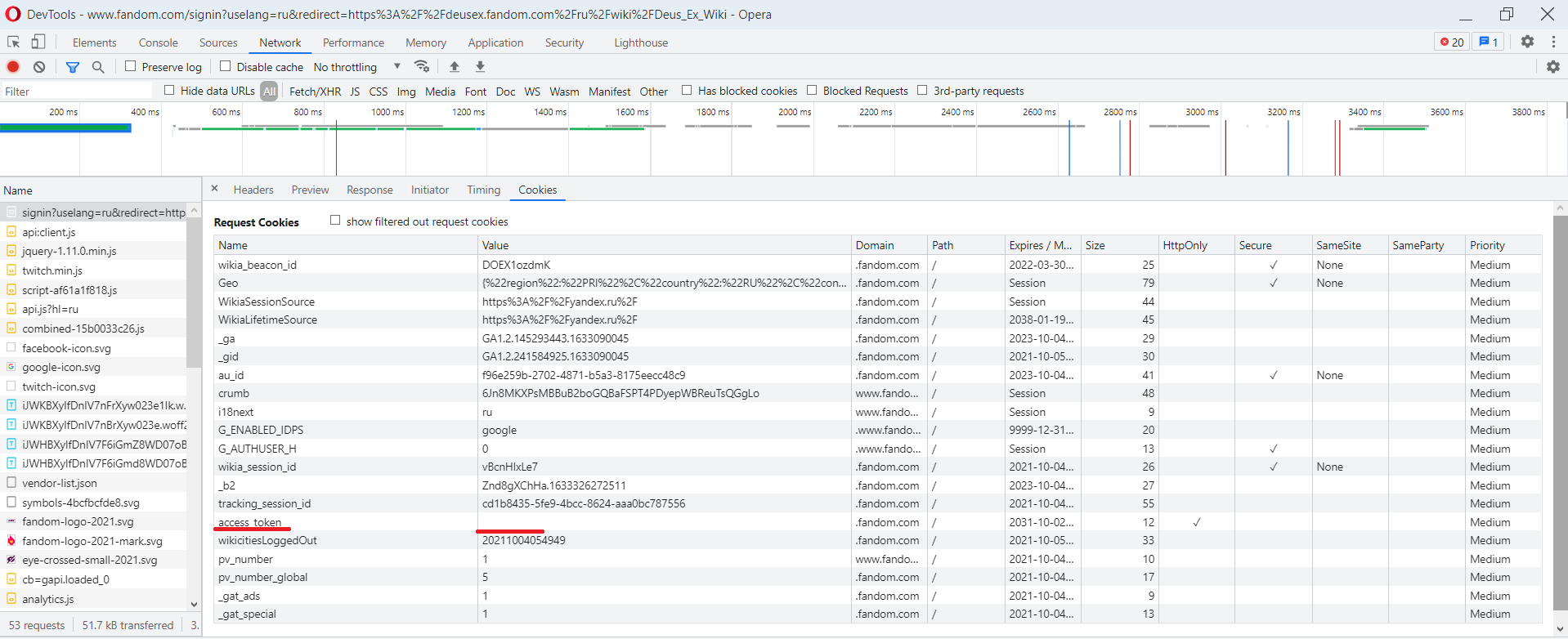
\*В обоих примерах в демонстрации ответа сервера красным подчёркнуты элементы cookie, в которых, предполагаемо, хранится ответ сервера.

### Сайт №1: DeusEx Wiki (fandom.com)

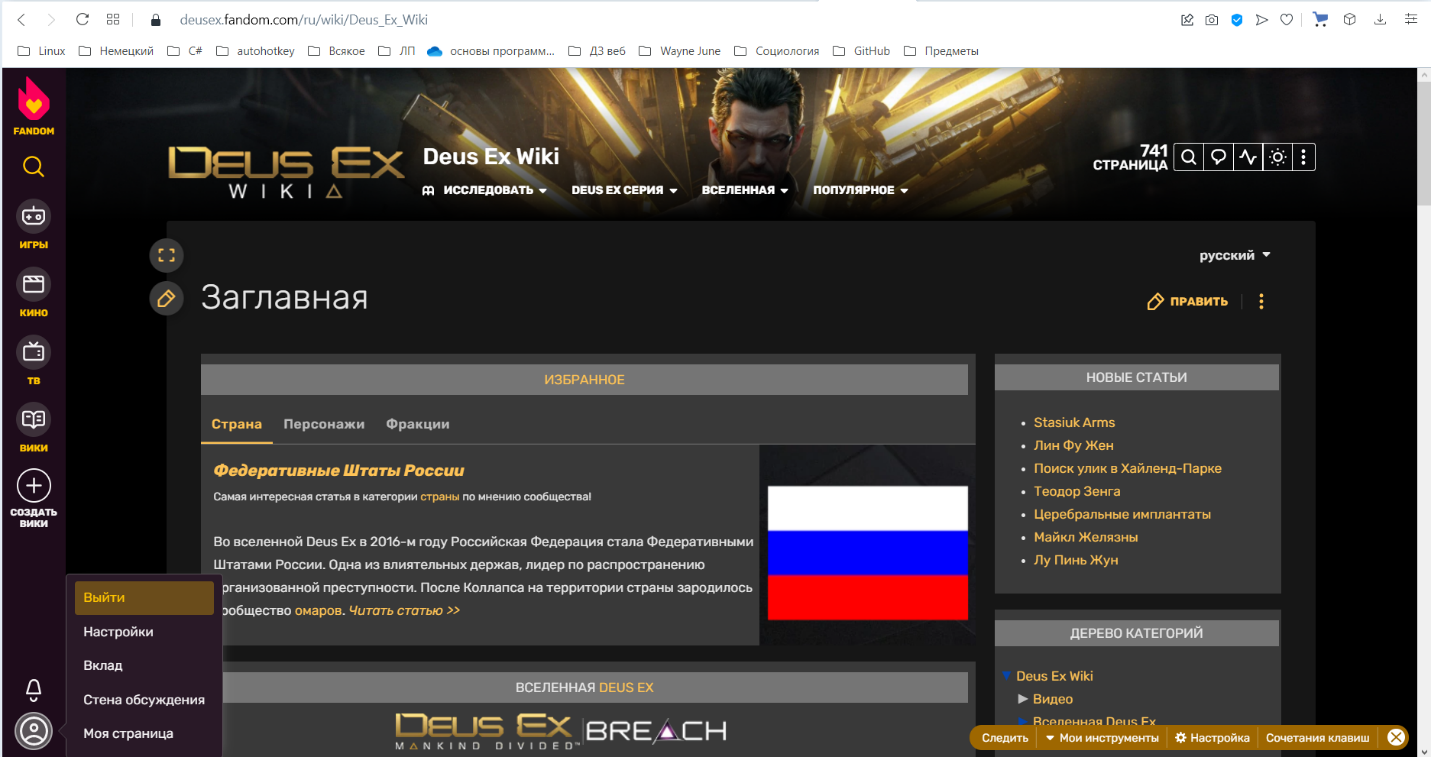
1) Отправка данных в форме авторизации



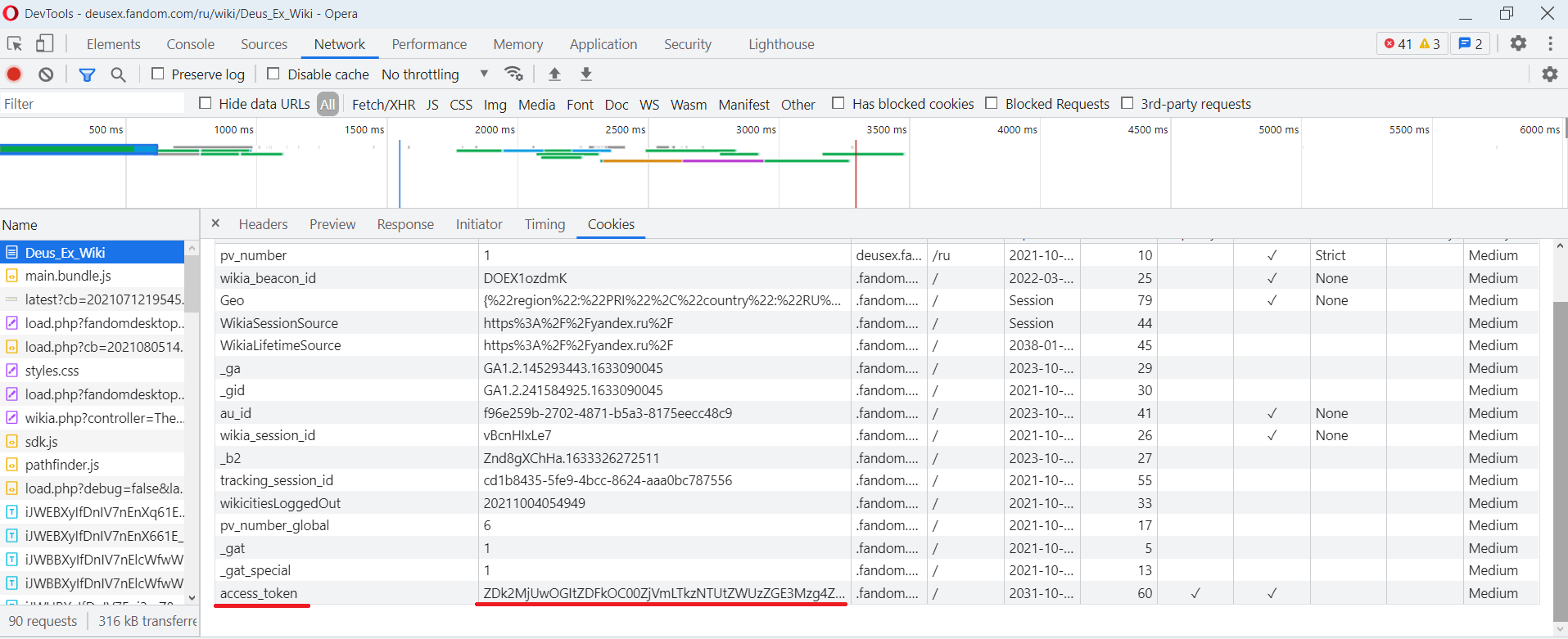
2) Файлы cookie до авторизации: поле *access\_token* пусто



3) Успешная авторизация



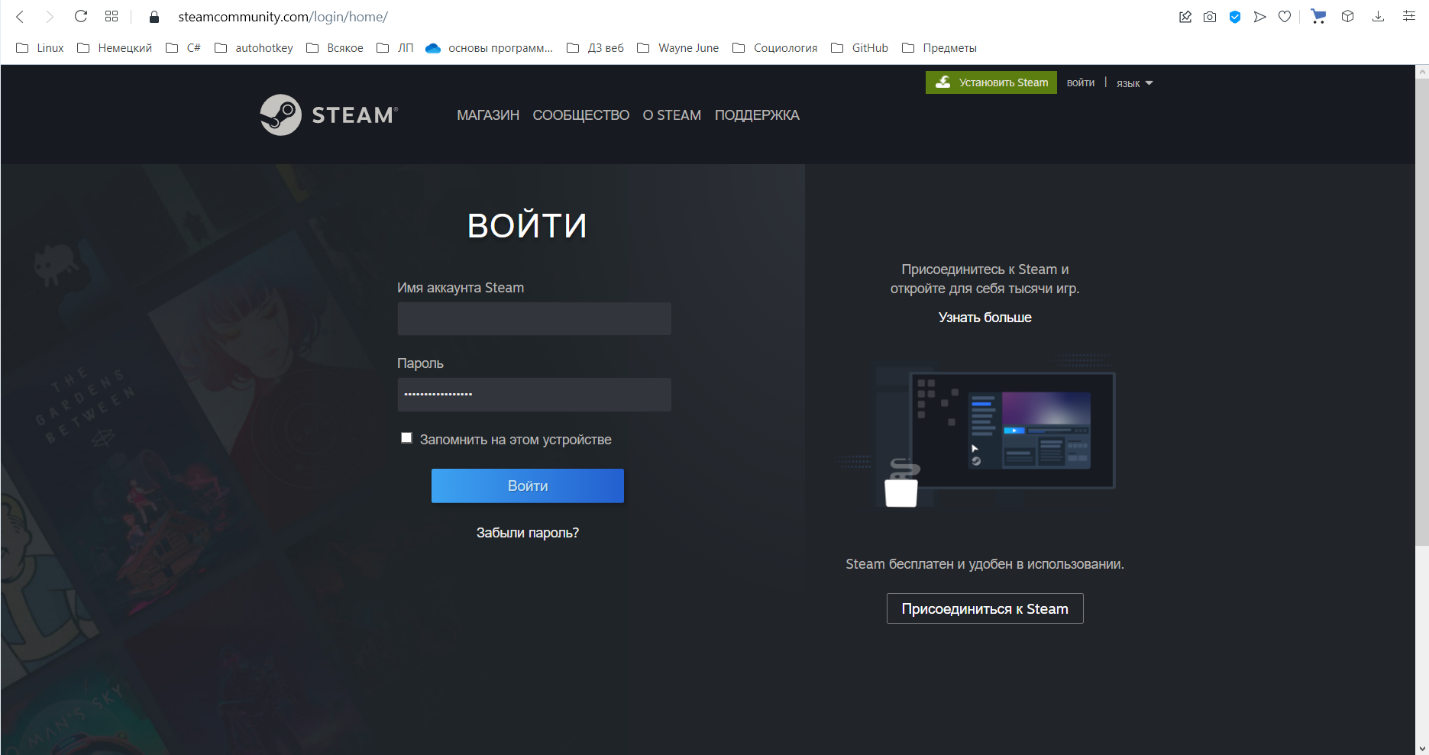
4) Файлы cookie после авторизации: поле *access\_token* обзавелось содержимым (все прочие поля неизменны и совпадают с аналогами до авторизации)



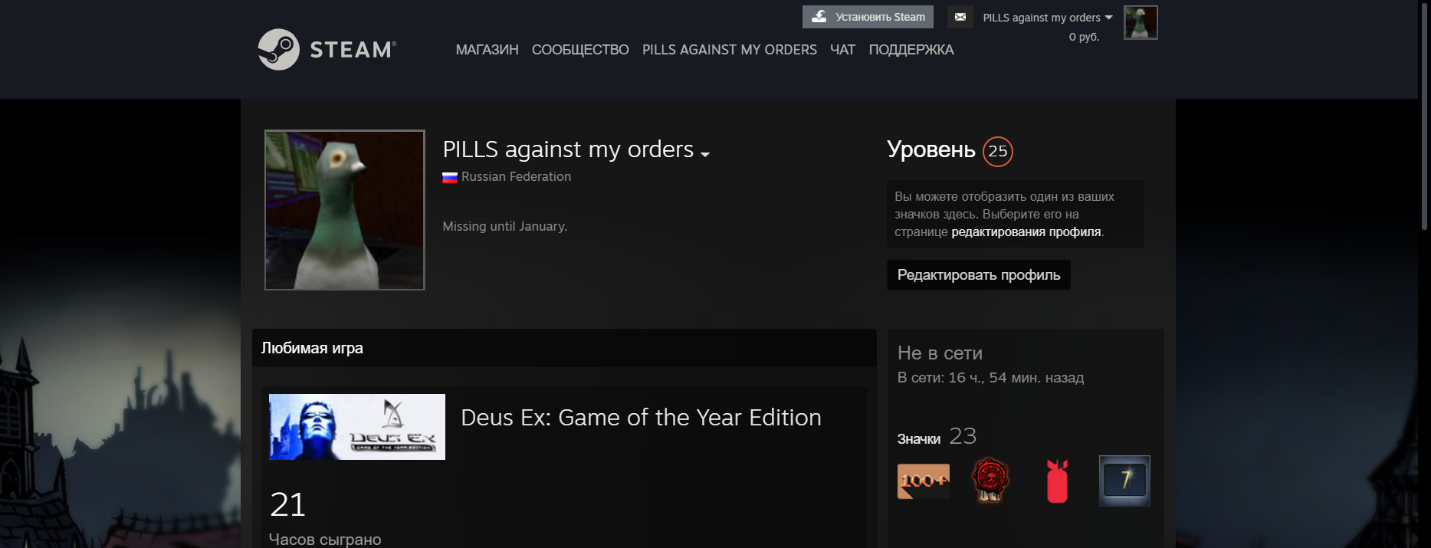
\*В данном примере данные до авторизации указаны для того, чтобы обосновать выбор поля, в котором хранится ответ сервера.

### Сайт №2: Steam (steamcommunity.com)

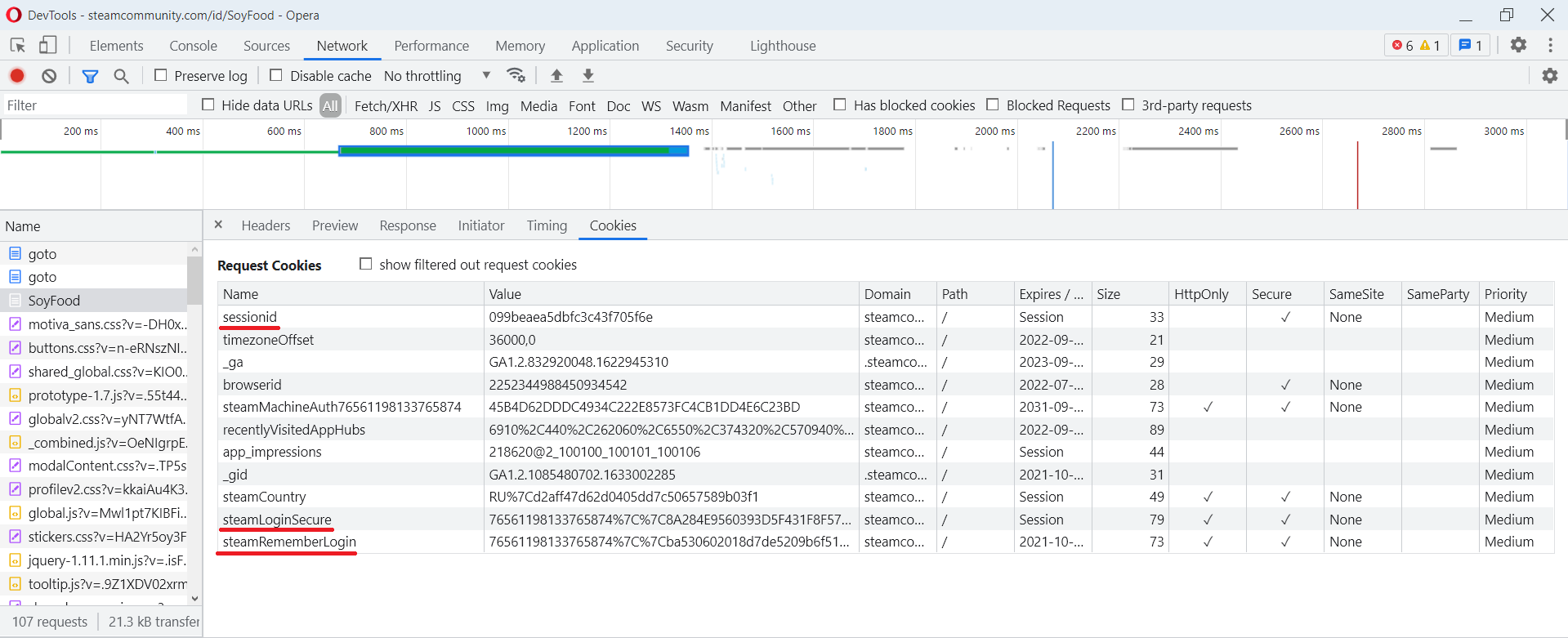
1) Отправка данных в форме авторизации



2) Успешная авторизация



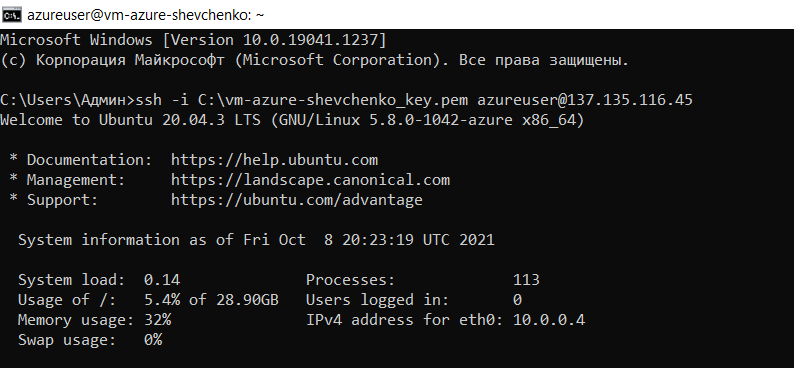
3) Файлы cookie после авторизации



## ДЗ №3: Установка VM (Microsoft Azure)

\*aka «Virtual Machine»

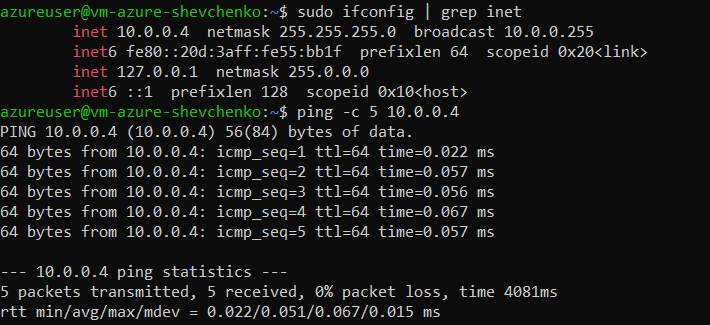
### 1) Успешный вход в ВМ на Azure



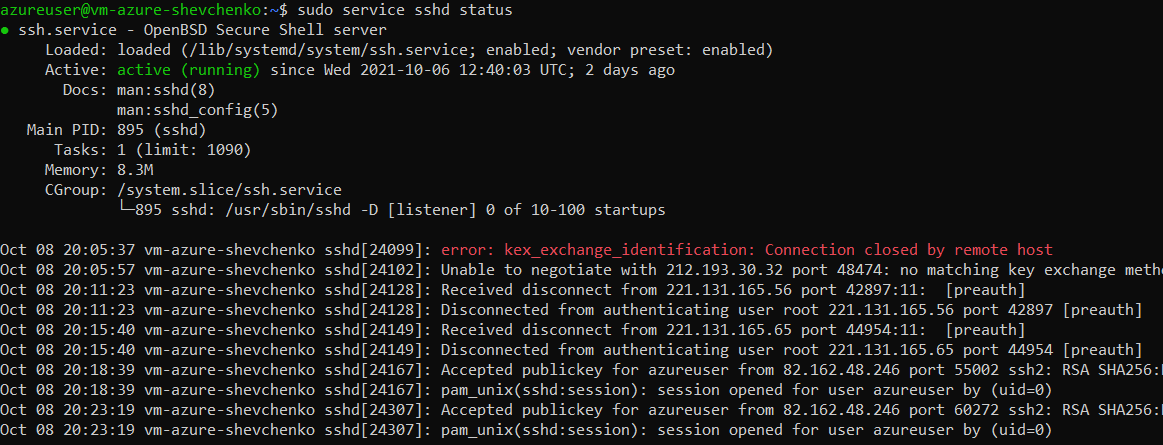
### 2) Обмен пакетами с google.com



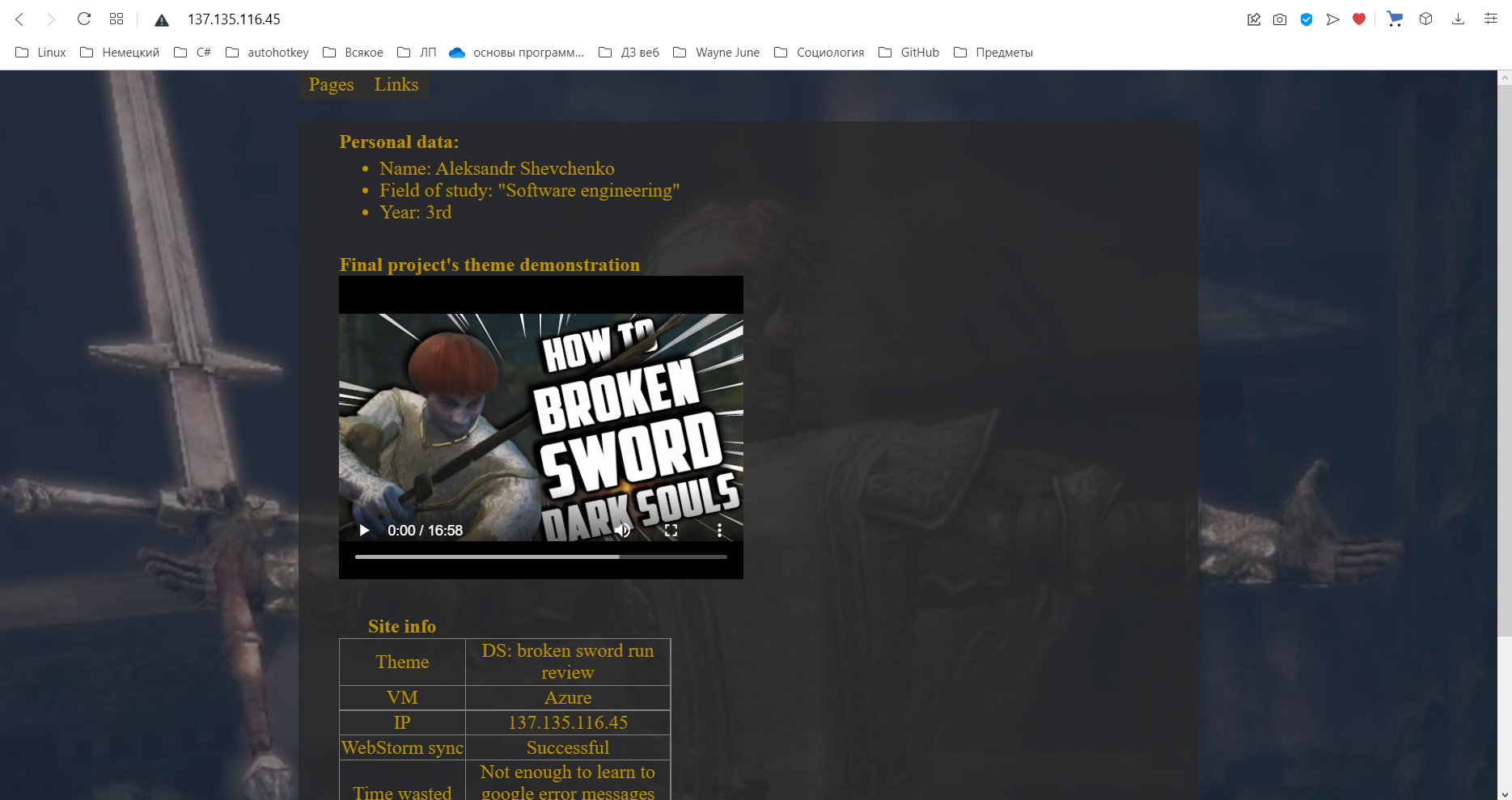
### 3) Обмен пакетами с сетевым интерфейсом



### 4) Проверка готовности сервера apache



### 5) Переход по ip-адресу сервера (на данном этапе уже прошли синхронизация с WS и «деплой» веб-страницы)



## ДЗ №4-5: Страничка «О себе» и Piano/DropMenu

\*aka «15 tags of me» & «Piano/DropMenu»

Адрес для проверки: http://137.135.116.45

Архив с сайтом (без видео): в репозитории (Final\_project.7z)

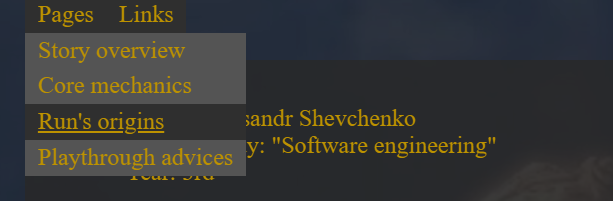
**Перечень использованных тегов**: <link>, <img>, <div>, <ul>, <ol>, <li>, <b>, <i>, <table>, <tr>, <td>, <th>, <video>, <a>, <style>.

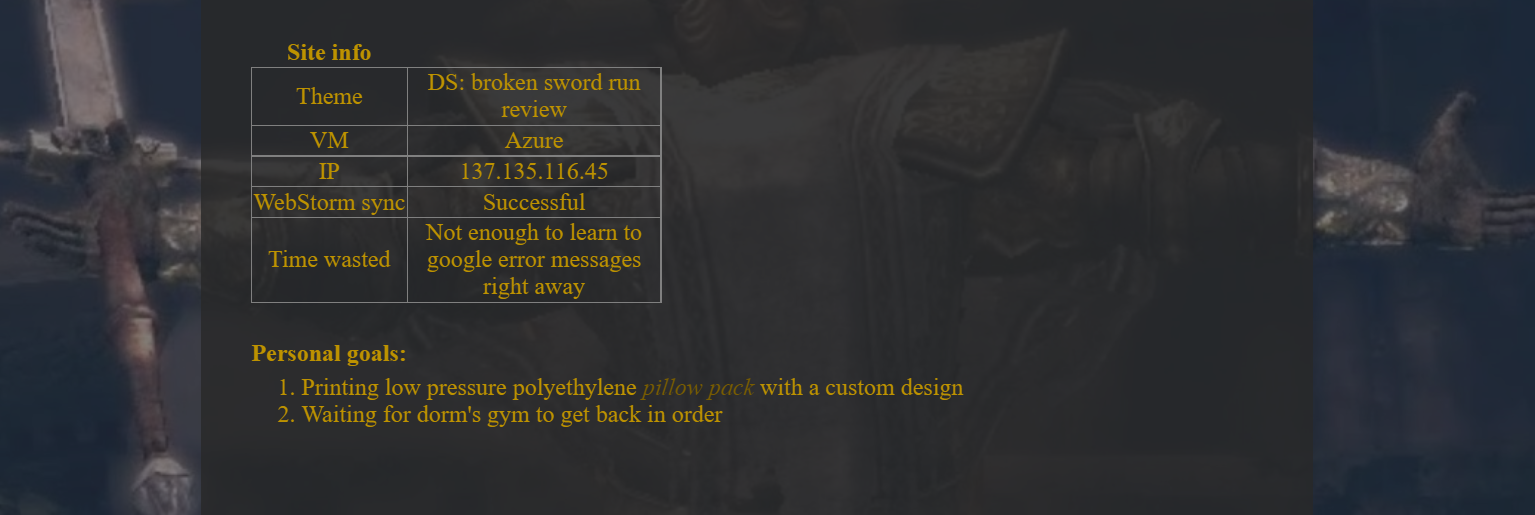
**Перечень использованных атрибутов**: class, id, alt, src, href, style.

**Реализованная панель навигации**: «в ряд», с выпадающими списками.

**Демонстрация**:





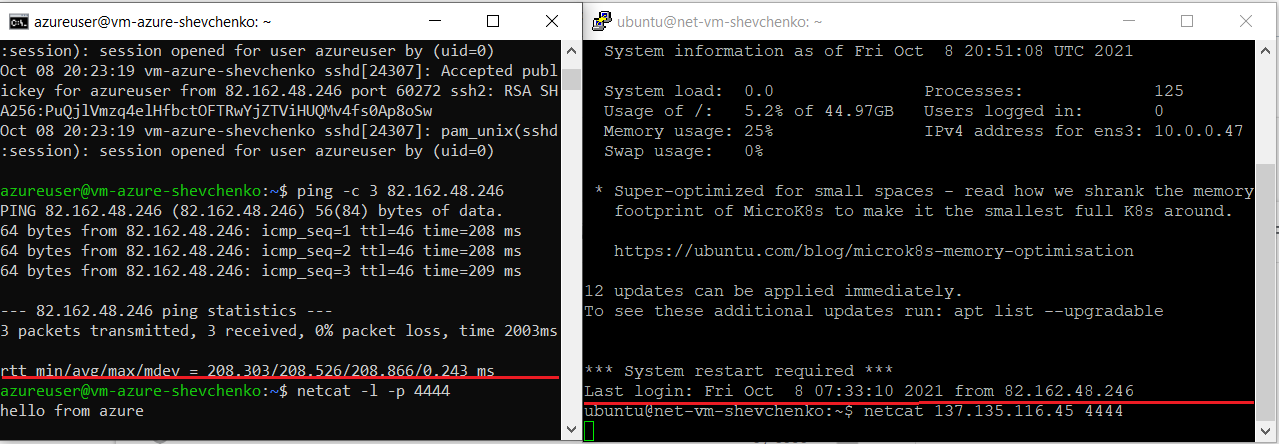


## ДЗ №6: Общение между серверами

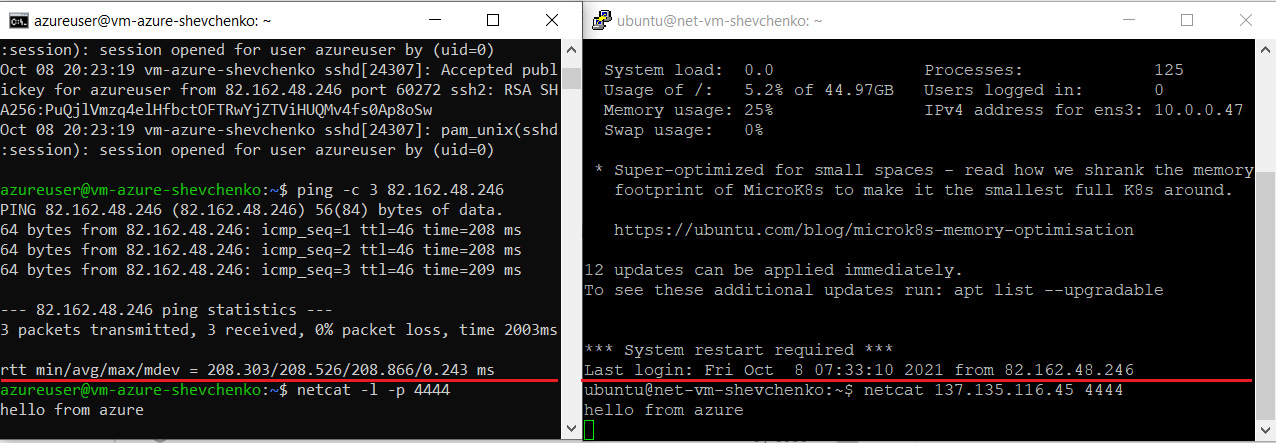
\*aka «Screens of Communications»

### 1) Связь через netcat

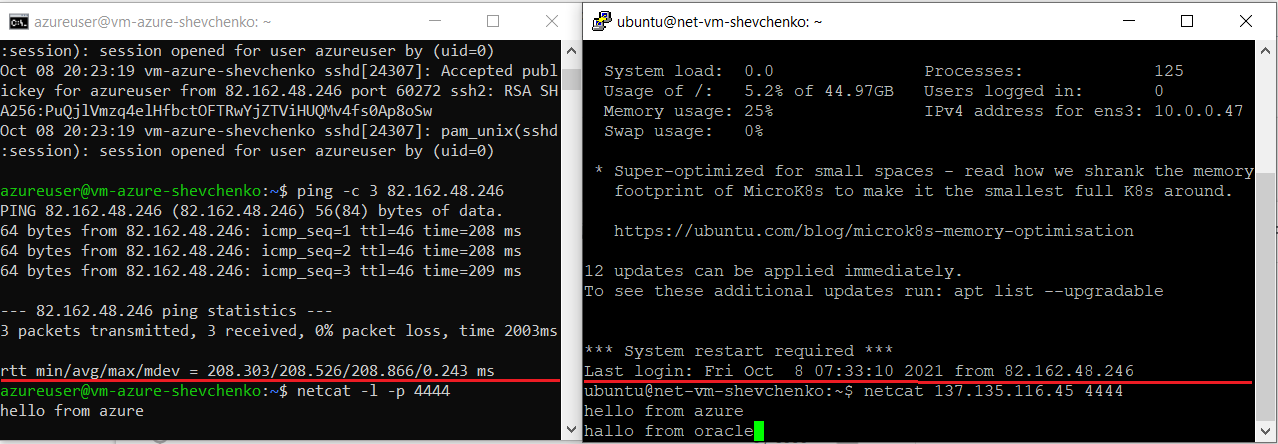
Установка соединения, отправка сообщения от VM-azure (слева)



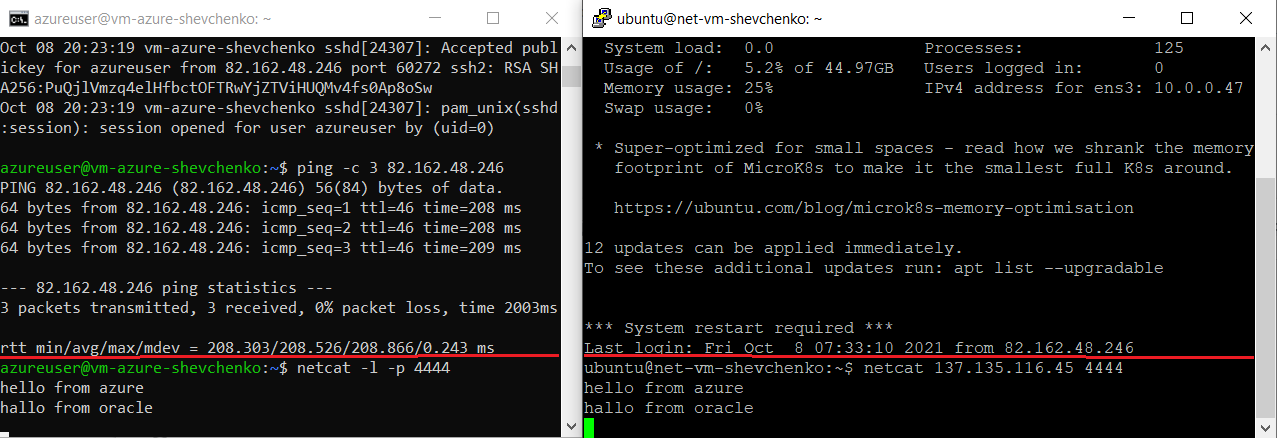
Отображение полученного сообщения на терминале VM-oracle (справа)



Отправка сообщения от VM-oracle

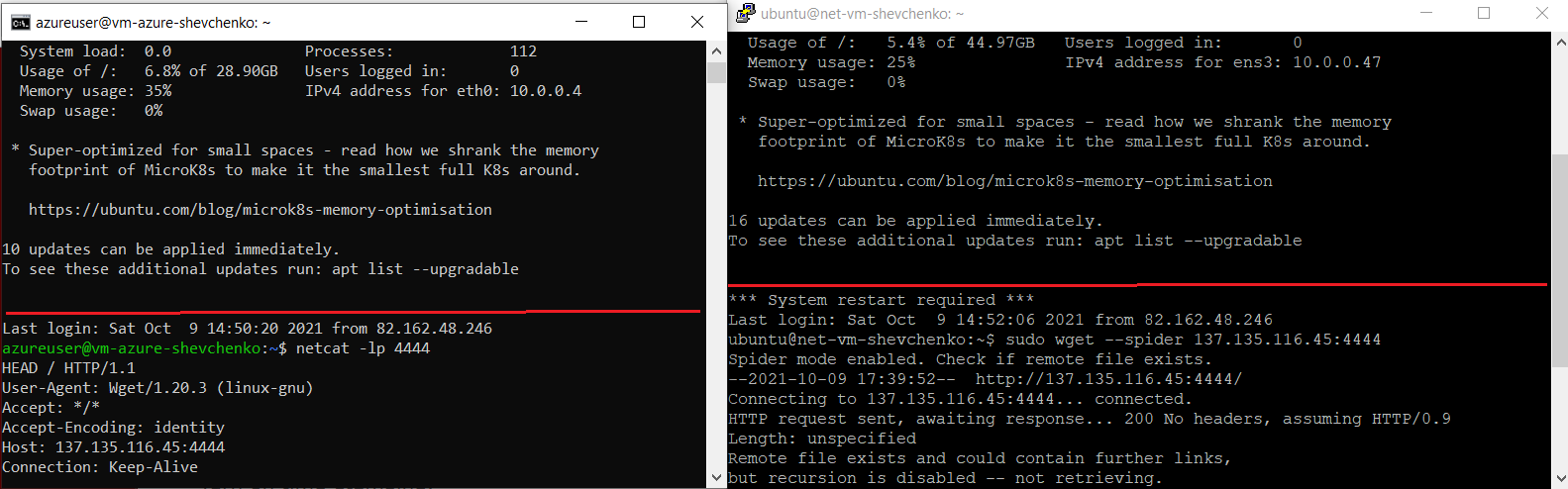


Отображение полученного сообщения на терминале VM-azure



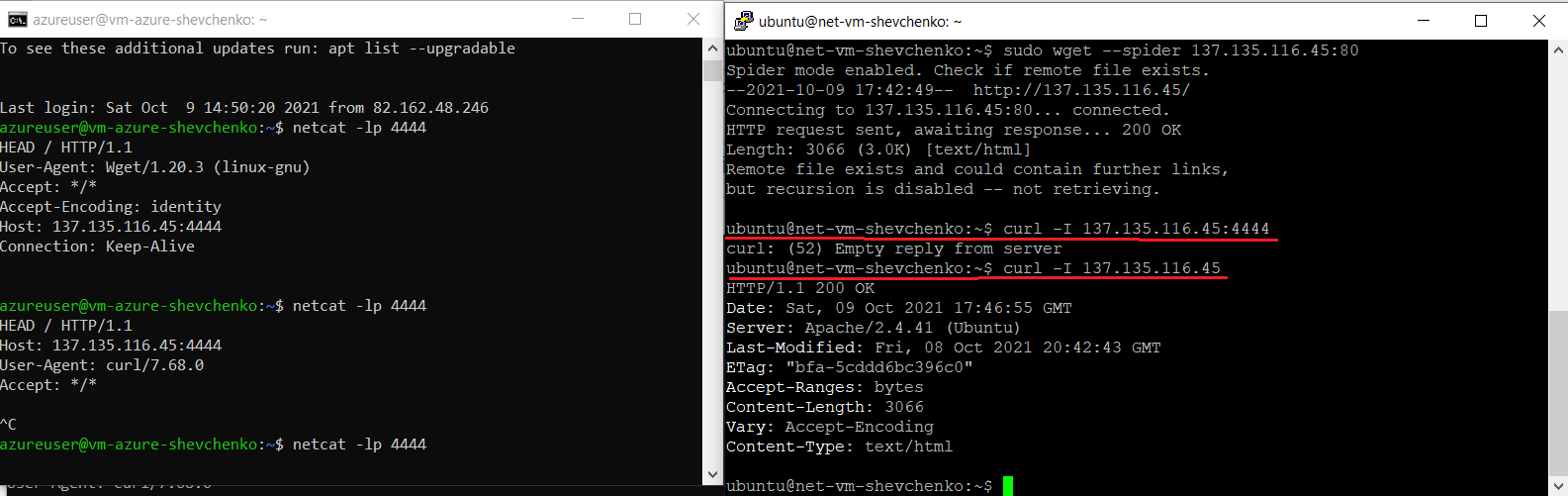
### 2) HTTP-запросы через wget

Хост получает запрос на прослушиваемый порт и, помимо отправки ответа (команда *–spider* запрашивает информацию о состоянии хоста, т.е. получен ответ «remote file exists»), выводит его у себя в полном виде. Почему именно в таком виде? Сам хотел бы знать.



### 3) HTTP-запросы через curl

Опция *–I* команды *curl* запрашивает у указанного хоста сетевые заголовки. Порт 4444 неспособен предоставить ответ, однако при обращении без указания порта (или с указанием 80) мы получаем нужный ответ.



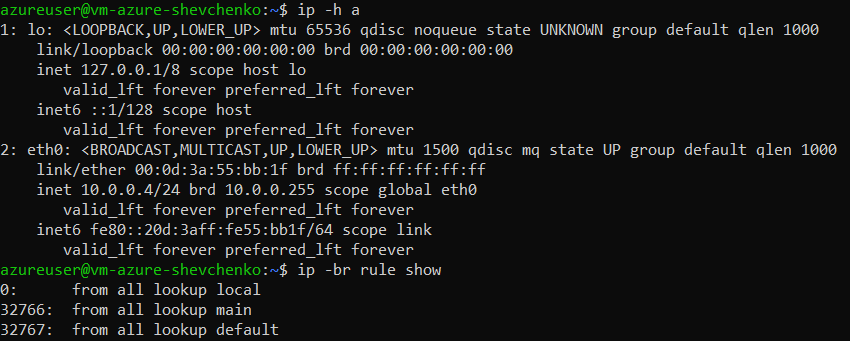
### 4) Отправка запросов через telnet

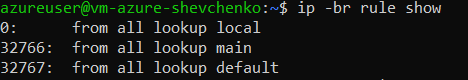
С помощью самой команды telnet устанавливаем соединение между клиентом и 80 портом оста 137.135.116.45. Дальше использована команда GET /, в ответ на которую хост вернул разметку страницы, расположенную по адресу.



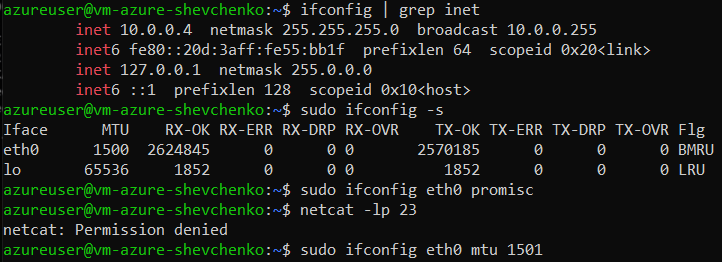
### Использование утилит

1. ip

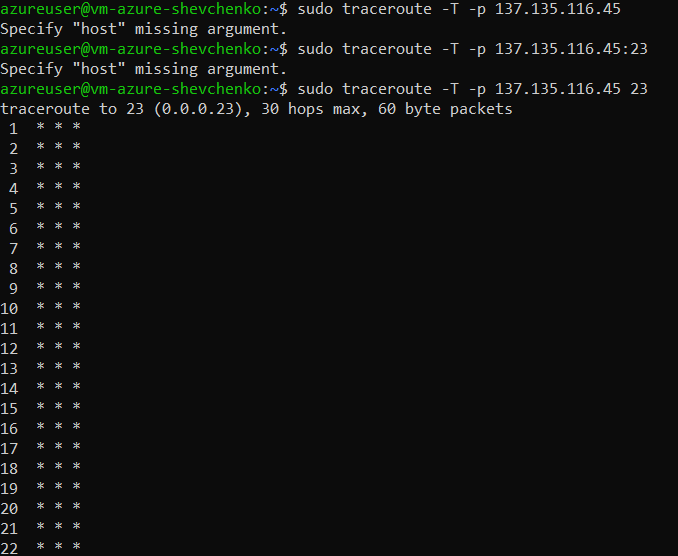


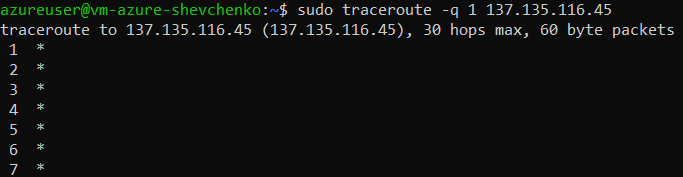


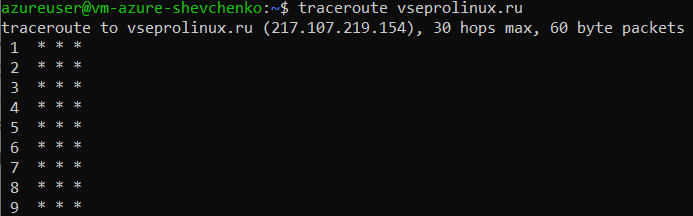
1. ifconfig



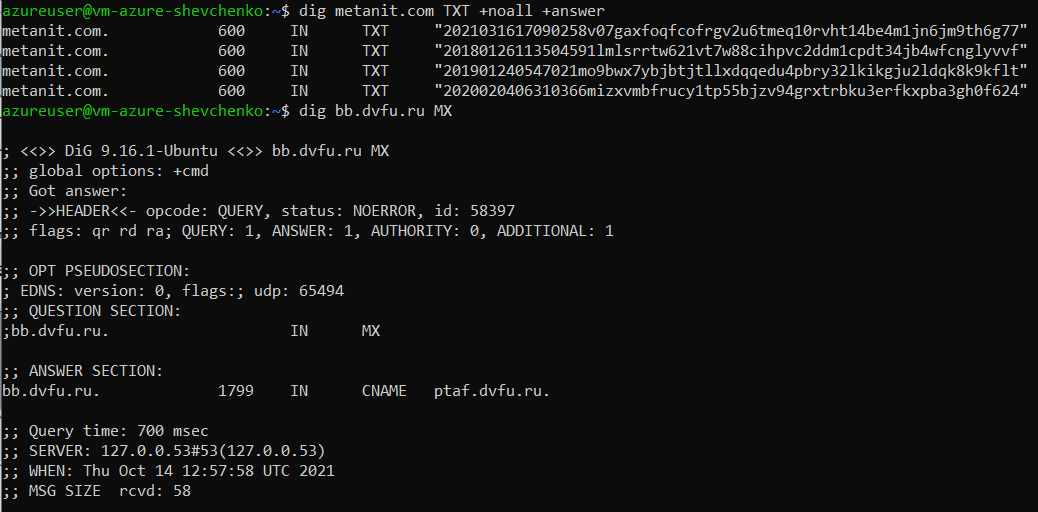
1. traceroute (не хотела сотрудничать)







1. dig



1. nslookup

