Исследование HTTP протокола

1. Найти в интернете 8 различных status code HTTTP. В запросе и ответе должно

содержаться не менее 5 атрибутов.

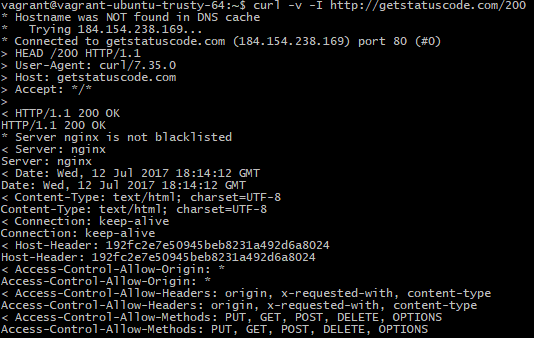


Рисунок 1.

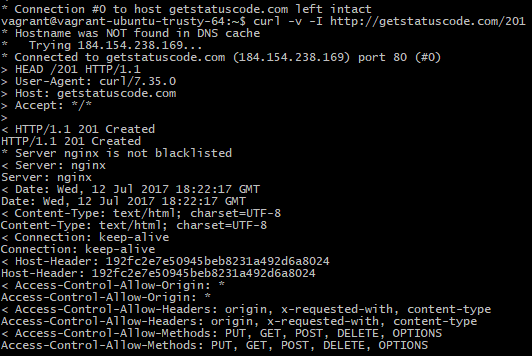


Рисунок 2.

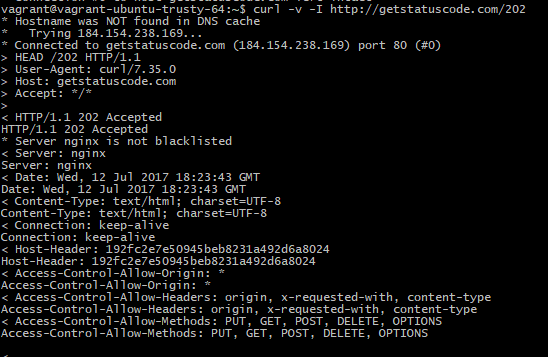


Рисунок 3.

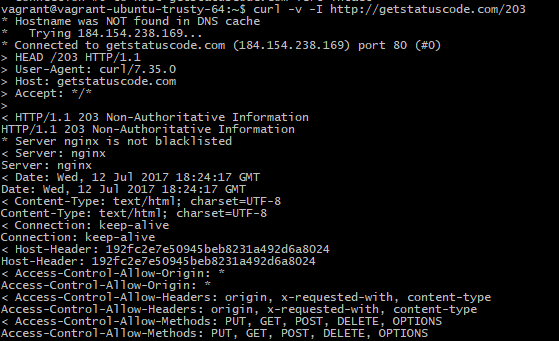


Рисунок 4.

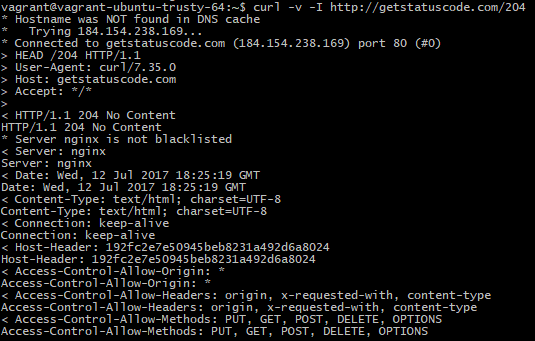


Рисунок 5.

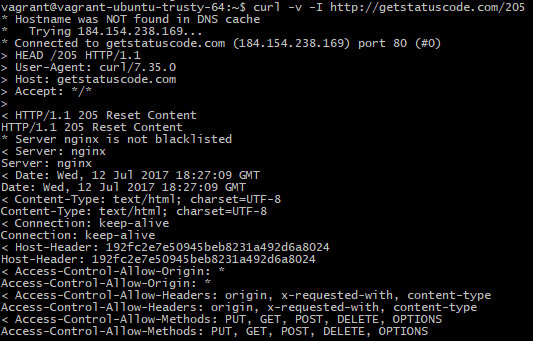


Рисунок 6.

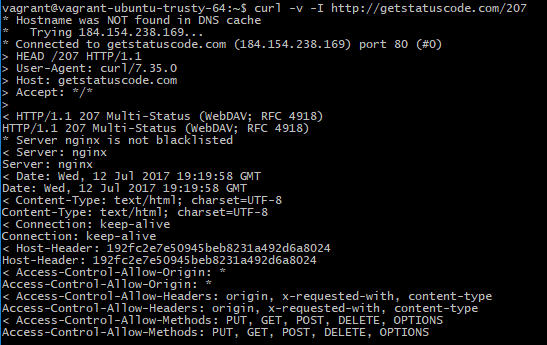


Рисунок 7.

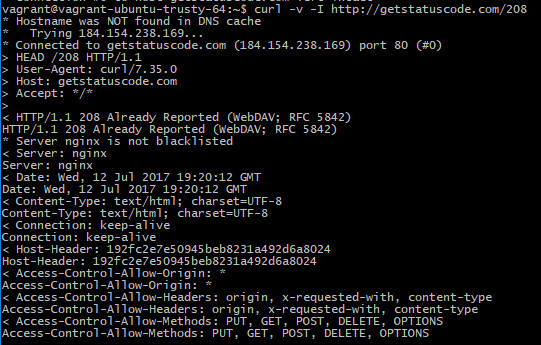


Рисунок 8.

2. Описать назначение всех атрибутов в client request and server response. На примере

одного из HTTP request/response описать все атрибуты

vagrant@vagrant-ubuntu-trusty-64:~$ curl -v -I http://getstatuscode.com/209

\* Hostname was NOT found in DNS cache

\* Trying 184.154.238.169...

\* Connected to getstatuscode.com (184.154.238.169) port 80 (#0)

ЗАПРОС

> HEAD /209 HTTP/1.1

Здесь используется метод HEAD для извлечения хедеров (метаданных) без содержания ресурса. Далее запрашиваемая страница и версия протокола HTTP.

> User-Agent: curl/7.35.0

Список названий и версий клиента и его компонентов с комментариями

> Host: getstatuscode.com

Доменное имя хоста.

> Accept: \*/\*

Информируем сервер о тех типах данных, которые нам нужны

>ОТВЕТ

< HTTP/1.1 302 Moved Temporarily

Версия протокола, статус код 302 и его значение. 302 Moved Temporarily — запрошенный документ временно доступен по другому URI.

< Server: nginx

Название веб-сервера.

< Date: Wed, 12 Jul 2017 19:20:25 GMT

Дата генерации отклика.

< Content-Type: text/html; charset=UTF-8

Формат и способ представления сущности.

< Connection: keep-alive

Данный хедер указывает на то, что клиент хотел бы держать соединение открытым.

< Location: /404

Этот хедер указывает на ту страницу на которую будет перенаправлен клиент.

< Host-Header: 192fc2e7e50945beb8231a492d6a8024

Пользовательский хедер сервера, несущий информацию полезную серверу.

< Access-Control-Allow-Origin: \*

Заголовок определяющий ресурсы, которым может быть отправлен ответ

< Access-Control-Allow-Headers: origin, x-requested-with, content-type

Заголовок определяющий поля, которые может использовать клиент при формировании запроса

< Access-Control-Allow-Methods: PUT, GET, POST, DELETE, OPTIONS

Заголовок определяющий методы, которыми клиент может пользоваться при формировании запроса

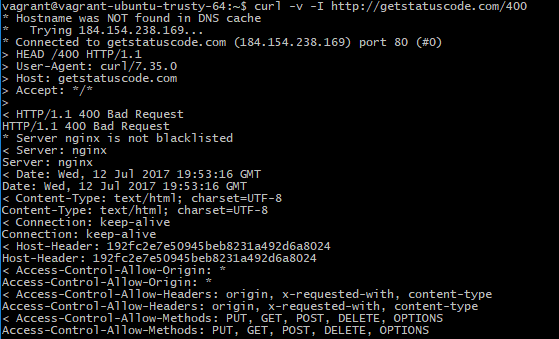
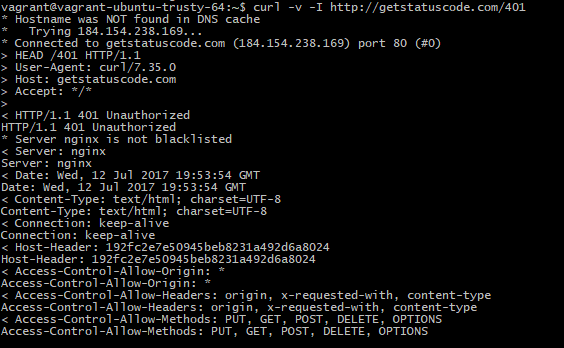
3. Найти еще 7 различных status code. Выполнять только после выполнения задания 1.

Рисунок 9.



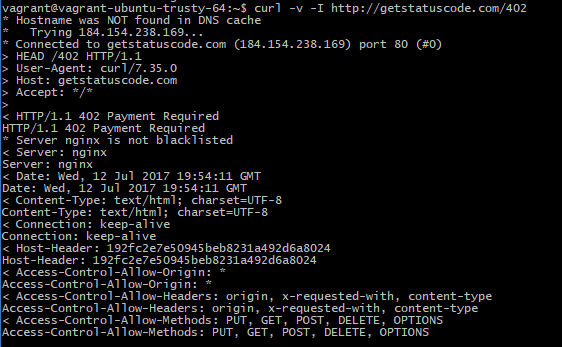
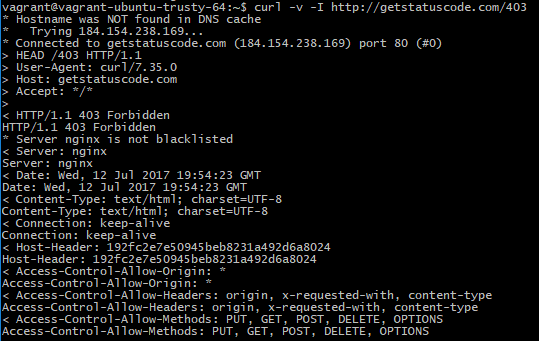
Рисунок 10.

Рисунок 11.



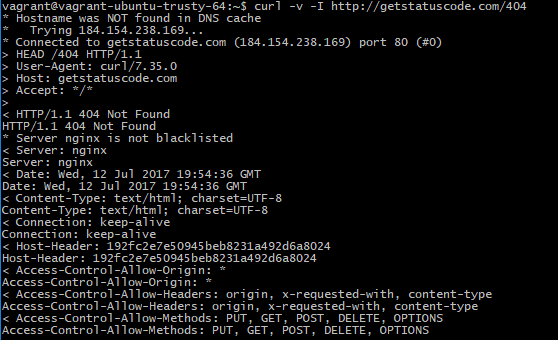
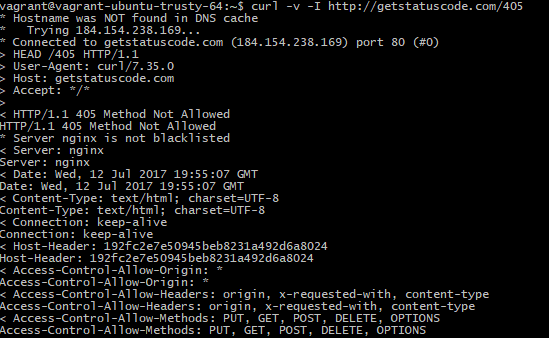
Рисунок 12.

Рисунок 13



.

Рисунок 14

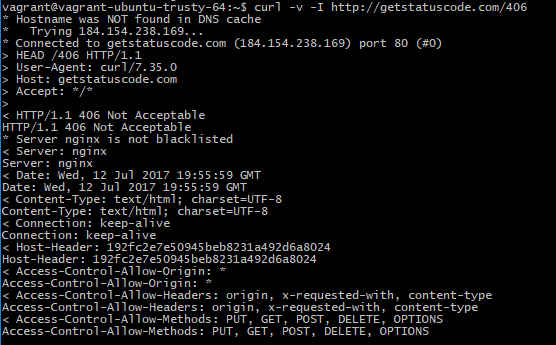
.

Рисунок 15.

4. Произвести фильтрацию трафика протокола HTTP с помощью tcpdump. Написать два

фильтра:

a. фильтровать по методам протокола HTTP

GET = 0x47455420

POST = 0x504f5354

HEAD = 0x48454144

port 80 and tcp[((tcp[12:1] & 0xf0) >> 2):4] = 0x47455420

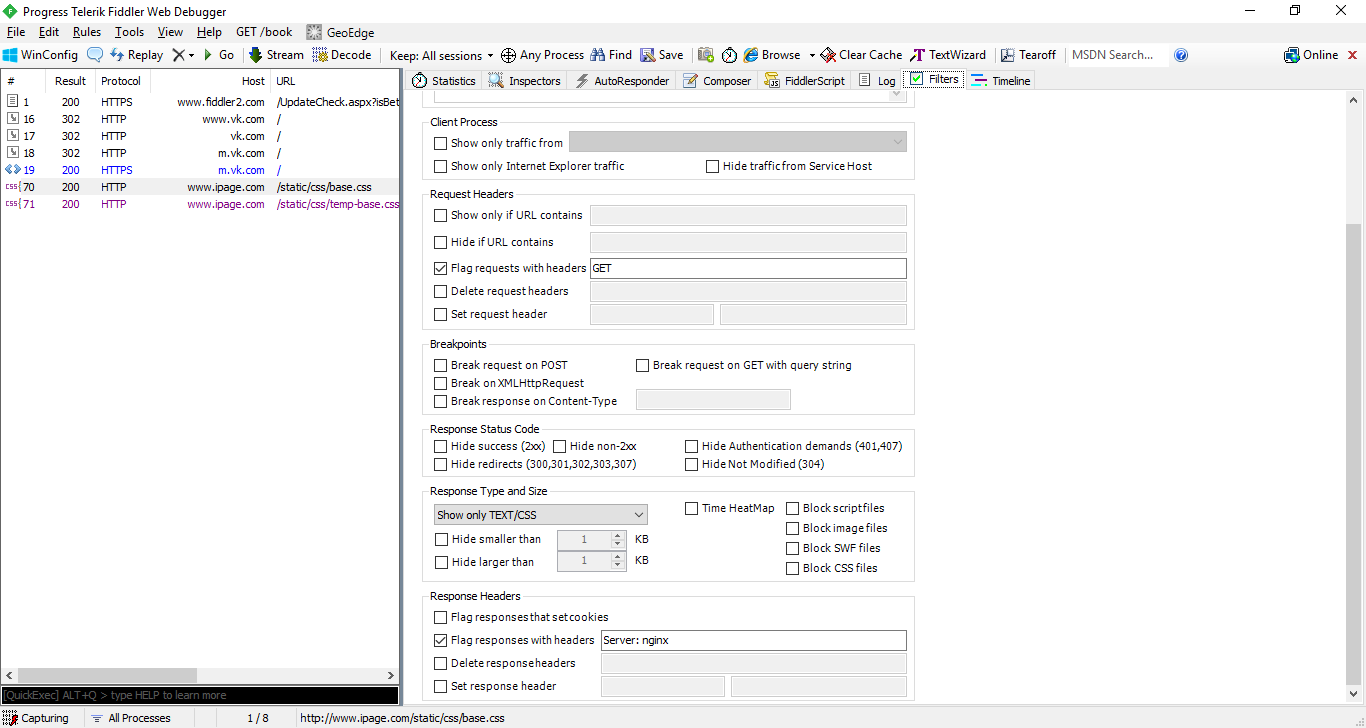
b. фильтровать по методу и атрибуту в response протокола HTTP

port 80 and tcp[((tcp[12:1] & 0xf0) >> 2):4] = 0x47455420 | grep –i “Server: nginx\*”

c. фильтровать по методу и атрибуту в request протокола HTTP

port 80 and tcp[((tcp[12:1] & 0xf0) >> 2):4] = 0x47455420 | grep –i “Server: nginx\*”

5. Используя Fiddler выполнить пункт 4.



Исследование API GitLab

Используя утилиту curl написать запросы:

• создание нового проекта,

• удаление проекта

• добавления пользователей с различными ролями

• создание issue и назначение его определенному пользователю

• получение списка пользователей

o весь

o с определенными правами

• работа с коммитами

o получить список всех комментариев коммита

o вставить комментарий в commit в определённую строку от имени

пользователя

\* Работа в группах по два, три человека