Уральский федеральный университет

имени первого Президента России Б. Н. Ельцина

Кафедра «Интеллектуальные информационные технологии»

**ЛАБОРАТОРНАЯ РАБОТА 3**

ОТЧЁТ

Дисциплина «Web-программирование»

Преподаватель:

Д.В. Свинцов

Студент:

гр. РИ-380022

Борисова В. Б.

Екатеринбург, 2020

**Лабораторная работа 3**

**Работа с протоколом HTTP через telnet**

1. Задание:

Отправить запросы на <http://httpbin.org>, проанализировать ответ и код состояния. Описать работу HTTP протокола в каждом запросе.

* Запросить данные GET запросом с ресурса ip
* Выполнить запрос методом GET
* Выполнить запрос методом POST

Попробовать ввести неверное значение Content-Length.

* Отправить запрос на установку Cookie
* Просмотреть список установленных Cookie
* Отправить запрос на страницу с перенаправлением

Проверить глубину рекурсии в браузере, сравниь со значением опции network.http.redirection-limit из about:config в браузере FireFox.

1. Ход работы:

Подключаемся по telnet к <http://httpbin.org> с помощью команды:

$ telnet httpbin.org 80

2.1. **Затем нужно запросить данные GET запросом с ресурса ip:**

GET /ip HTTP/1.1 #GET запрос по протоколу HTTP 1.1 для ip

Host: httpbin.org # отправка запроса на хост

Accept: \*/\* # принимает любые типы данных

(Рисунок 1)



Рисунок 1 — Результат выполнения запроса

Запрос был выполнен с кодом 200, то есть обработан успешно. Так же запрос вернул информацию о внешнем ip-адресе.

2.2 **Дальше выполним запрос методом GET:**

GET /get?foo=bar&1=2&2/0&error=True HTTP/1.1 # GET запрос с методом get, который на вход получает набор аргументов

Host: httpbin.org # хост, с которым связывается GET запрос

Accept: \*/\* # принимает любые типы данных

(Рисунок 2)

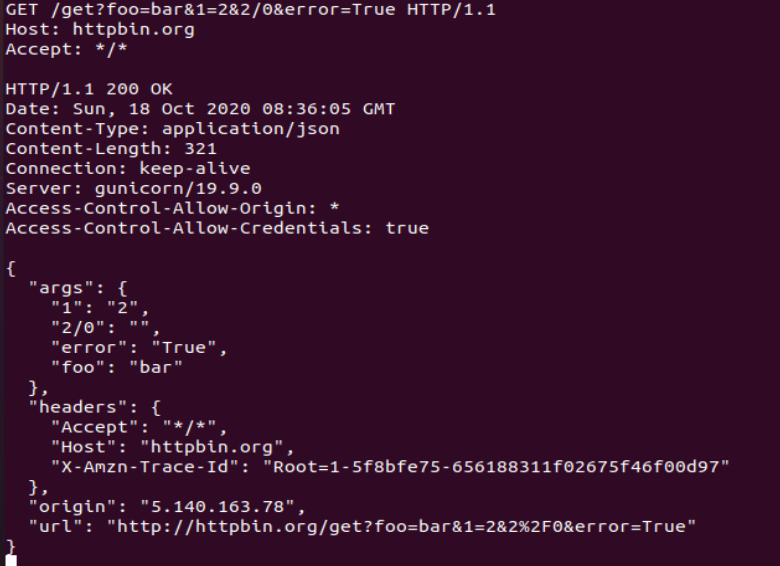


Рисунок 2 — Результат выполнения запроса GET

Запрос был выполнен с кодом 200, то есть обработан успешно. Так же запрос вернул информацию об аргументах, переданные методу get.

2.3 **Запрос методом POST:**

POST /post HTTP/1.1

Host:httpbin.org

Accept: \*/\*

Content-Length: 29 # число = длине передаваемого контента

Content-Type: application/x-www-form-urlencoded # тип отправляемых данных

foo=bar&1=2&2%2F0=&error=True

(Рисунок 3)

****

Рисунок 3 — Результат выполнения POST-запроса

Запрос был выполнен с кодом 200, то есть обработан успешно. Переданная информация отразилась в поле “form”.

Попробуем передать неверное значение Content-Length, видим, что часть переданных данных не передалась в поле “form”. (Рисунок 4)

****

Рисунок 4 — Передача неверного значения

2.4 **Отправка запроса на установку Cookie:**

GET /cookies/set?country=Ru HTTP/1.1

Host: httpbin.org

Accept: \*/\*

(Рисунок 5)

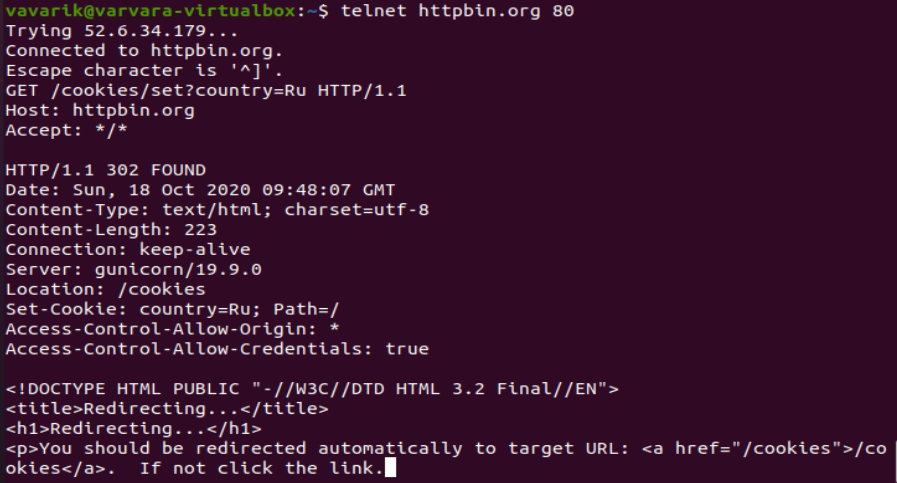


Рисунок 5 — Выполнение запроса на установку Cookie

Результатом выполнения запроса является перенаправление на другую страницу.

2.5 **Дальше проверим список установленных Cookie:**

GET /cookies HTTP/1.1

Host: httpbin.org

Accept: \*/\*

(Рисунок 6)

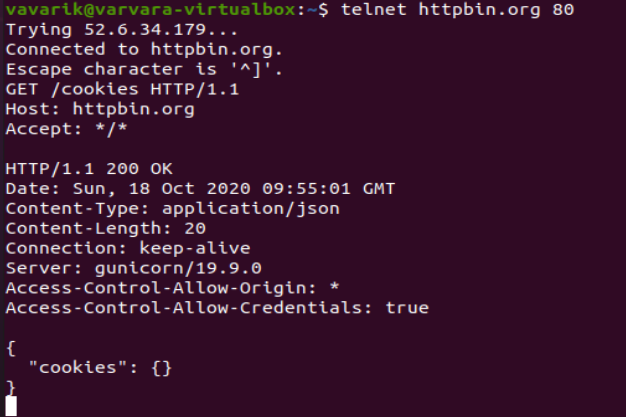


Рисунок 6 — Проверка установленных Cookie

Поле “cookies”: {} – пустое, значит на данный момент cookie не установлены.

2.6 **Отправить запрос на страницу с перенаправлением:**

GET /redirect/4 HTTP/1.1

Host: httpbin.org

Accept: \*/\*

(Рисунок 7)

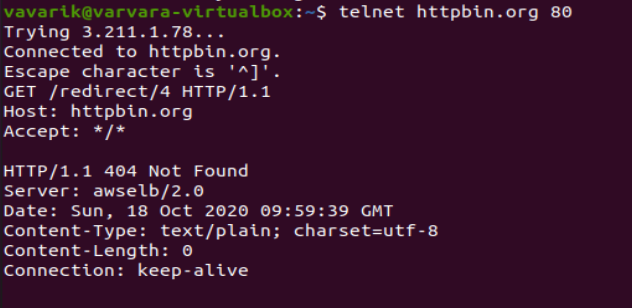
****

Рисунок 7 — Выполнение запроса на страницу с перенаправлением

Ошибка 404 говорит о том, что указанный в запросе ресурс не найден.

Глубина рекурсии 20 (Рисунок 8):



Рисунок 8 — Проверка глубины рекурсии