ПРАКТИЧЕСКАЯ РАБОТА 2. АНАЛИЗ ПРОТОКОЛА HTTP

Цели работы

* Освоить фундаментальные принципы протокола HTTP. Это включает понимание структуры HTTP-запросов и ответов, а также назначения и применения его ключевых методов, таких как GET, POST, PUT, DELETE.
* Сформировать практические навыки захвата и анализа сетевого трафика с помощью инструмента Wireshark.
* Научиться выявлять и анализировать HTTP-запросы и ответы в захваченных сетевых данных, определяя их ключевые компоненты.
* Изучить систему статус-кодов HTTP, понять их классификацию и смысл наиболее распространённых кодов ответов сервера.
* Исследовать назначение и функции HTTP-заголовков, проанализировать, как различные заголовки (например, Cache-Control, Content-Type, Cookie) управляют взаимодействием между клиентом и сервером и влияют на обработку данных.

Задание 1. Загрузка файла трассировки http

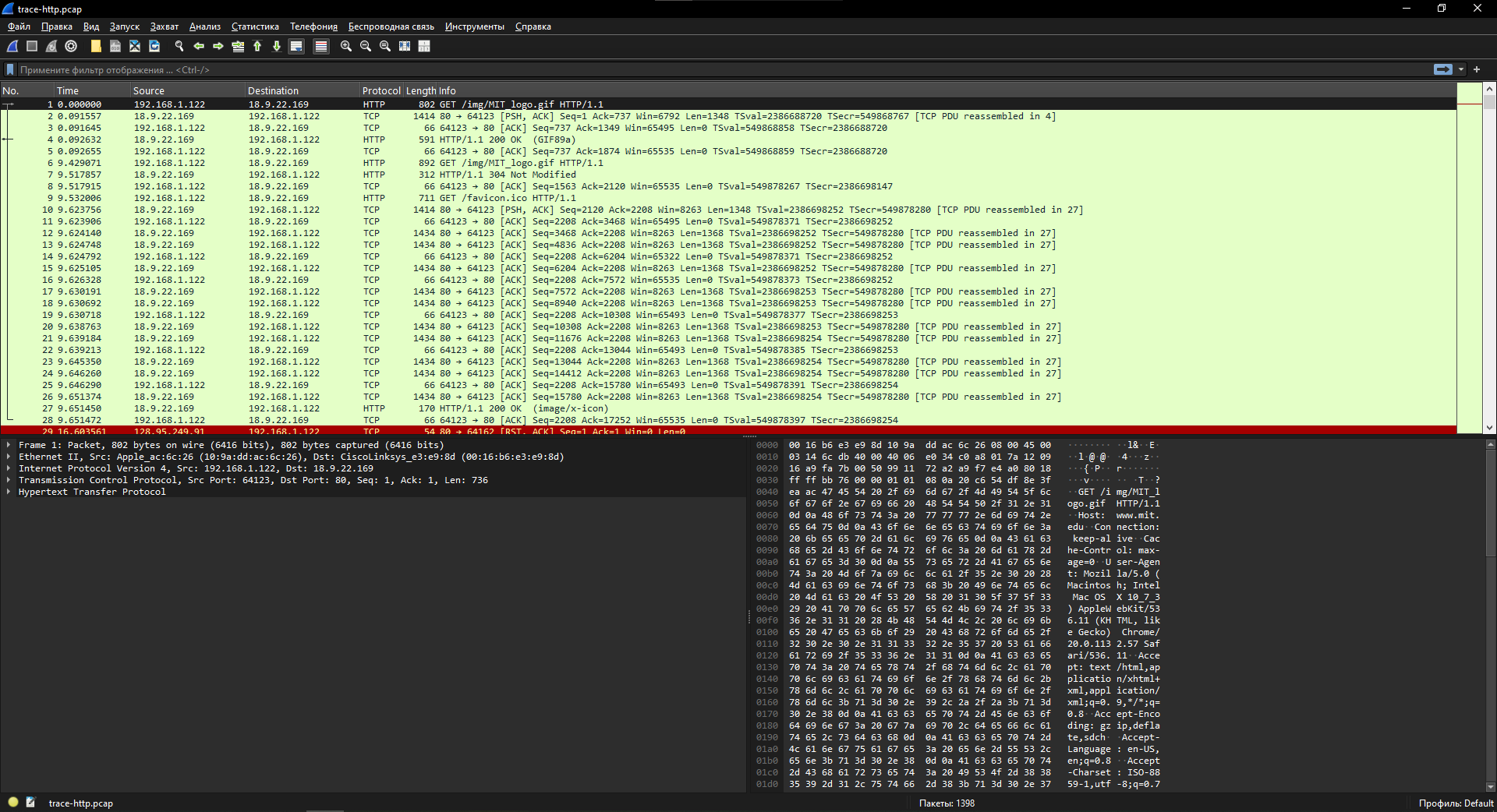


Рис. 1 открытие файла трассировки trace-http.pcap

• Количество пакетов, который были перехвачены Wireshark и присутствуют в этом файле? Ответ : 1398

Каждый пакет имеет временную метку и данные о кадре, которые говорят нам о времени создания пакета и о том, в каком именно кадре он передается. Другие данные включают адрес источника и назначения.

• Какой общий объем перехваченных пакетов (в байтах)? Ответ : 1223530

• Количество пакетов HTTP Request? Ответ : 98

• Количество пакетов HTTP Response? Ответ : 98

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 2 Проверка статистики пакетов

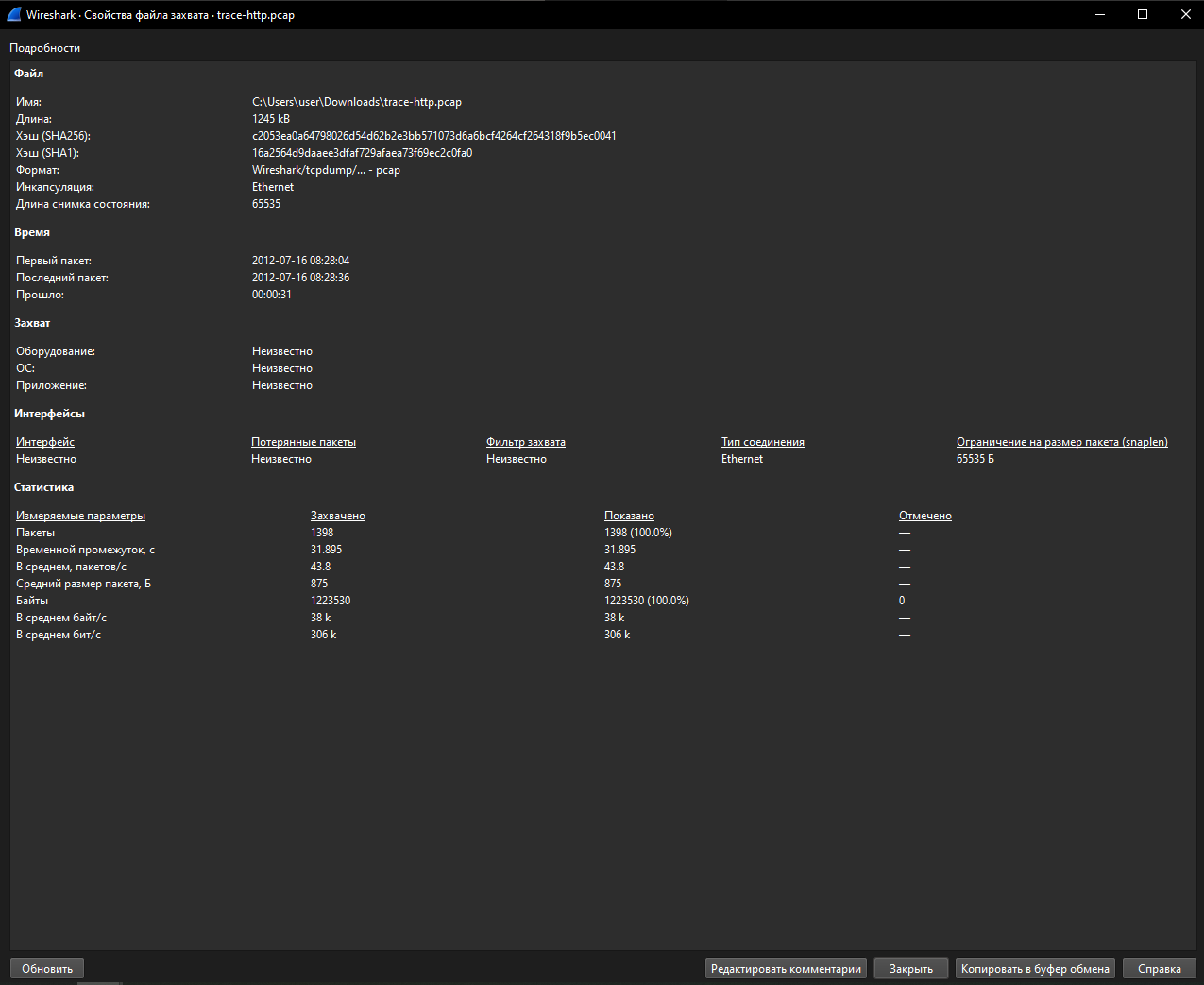


Рис. 3 Общая статистика сохранённого файла

Задание 2. Изучение трассировки запроса и ответа

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Применение фильтра для HTTP

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Содержимое, созданное искусственным интеллектом, может быть неверным.

TCP

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

TCP или UDP с использованием порта 80

Просмотр GET запроса и изучение HTTP-заголовка

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

первый GET в трассировке и разверните его HTTP блок (это будет 802 GET /img/MIT\_logo.gif HTTP/1.1), как показано ниже).

Host: [www.mit.edu\r\n](http://www.mit.edu/r/n)

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_7\_3) AppleWebKit/536.11 (KHTML, like Gecko) Chrome/20.0.1132.57 Safari/536.11\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8\r\n

Accept-Encoding: gzip,deflate,sdch\r\n

Accept-Language: en-US,en;q=0.8\r\n

[…] Cookie: \_\_gads=ID=8ad65dc86381f8d7:T=1303597795:S=ALNI\_MYTf6iFAaVdjJxmfT\_V1\_01Q1AyCg; \_\_utma=242276382.1429768304.1303597797.1329108360.1342416347.7; \_\_utmb=242276382.2.10.1342416347; \_\_utmc=242276382; \_\_utmz=242276382.1317333779.5.6

Cache-Control: max-age=0\r\n

Заголовки имеются

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изучение GET-запроса

Server: Apache/1.3.41 (Unix) mod\_ssl/2.8.31 OpenSSL/0.9.8j\r\n

Date: Mon, 16 Jul 2012 05:28:04 GMT\r\n

Last-Modified: Mon, 16 Jul 2012 04:00:09 GMT\r\n

Cache-Control: max-age=81125\r\n

Expires: Tue, 17 Jul 2012 04:00:09 GMT\r\n

ETag: "10eb008a-5f2-500391c9"\r\n

Вот подробный анализ HTTP-заголовков из нашего примера, а также объяснение, какую полезную информацию они содержат и что она означает.

1. Server

Пример: Server: Apache/1.3.41 (Unix) mod\_ssl/2.8.31 OpenSSL/0.9.8j

Описание: Этот заголовок указывает, какой веб-сервер обработал запрос. В данном случае:

Apache/1.3.41 — версия веб-сервера Apache.

Unix — операционная система, на которой работает сервер.

mod\_ssl/2.8.31 — модуль для поддержки SSL/TLS.

OpenSSL/0.9.8j — версия библиотеки OpenSSL, используемой для шифрования.

Полезная информация:

Позволяет определить, какое программное обеспечение используется на сервере.

Может быть полезно для диагностики совместимости или уязвимостей (например, если версия Apache или OpenSSL устарела).

2. Date

Пример: Date: Mon, 16 Jul 2012 05:28:04 GMT

Описание: Указывает дату и время, когда сервер сформировал и отправил ответ на запрос.

Полезная информация:

Помогает клиенту понять, насколько актуален ответ.

Используется для синхронизации времени между клиентом и сервером.

3. Last-Modified

Пример: Last-Modified: Mon, 16 Jul 2012 04:00:09 GMT

Описание: Указывает дату и время последнего изменения запрошенного ресурса (например, HTML-страницы, изображения).

Полезная информация:

Клиент может использовать этот заголовок для кэширования: если ресурс не изменился, сервер может вернуть ответ 304 Not Modified, экономя трафик.

Помогает определить, нужно ли загружать ресурс заново.

4. Cache-Control

Пример: Cache-Control: max-age=81125

Описание: Управляет кэшированием ответа. В данном случае:

max-age=81125 — ресурс можно кэшировать на 81125 секунд (примерно 22.5 часа).

Полезная информация:

Определяет, как долго клиент или промежуточные прокси могут хранить ответ в кэше.

Помогает снизить нагрузку на сервер и ускорить загрузку страниц для пользователей.

5. Expires

Пример: Expires: Tue, 17 Jul 2012 04:00:09 GMT

Описание: Указывает дату и время, после которого кэшированный ответ считается устаревшим.

Полезная информация:

Альтернативный способ управления кэшированием (часто используется вместе с Cache-Control).

Клиент не будет использовать кэшированный ответ после указанной даты.

6. ETag

Пример: ETag: "10eb008a-5f2-500391c9"

Описание: Уникальный идентификатор версии ресурса. Используется для проверки, изменился ли ресурс с момента последнего запроса.

Полезная информация:

Клиент может отправить этот идентификатор в следующем запросе (If-None-Match), чтобы сервер вернул ресурс только в случае его изменения.

Помогает избежать повторной передачи неизменённых данных.

Пример использования заголовков в GET-запросе

Представим, что клиент отправляет GET-запрос на страницу example.com/page.html.

Сервер возвращает ответ с указанными заголовками.

Клиент сохраняет ресурс в кэш и запоминает ETag и Last-Modified.

При следующем запросе клиент может отправить заголовки If-None-Match и If-Modified-Since.

Если ресурс не изменился, сервер вернёт 304 Not Modified, и клиент использует кэшированную версию, экономя трафик и время.

Задание 3. Кэширование контента

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

892 GET /img/MIT\_logo.gif HTTP 1.1

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

«If Modified-Since»

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

«304 Not Modified»



Сайт с Gif изображенем

IMG_256

Gif изображения

Задание 4. Сложные страницы

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

750 GET / HTTP/1.1

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

В разделах Статистика и HTTP

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Содержимое, созданное искусственным интеллектом, может быть неверным.

Отображение статистики по HTTP

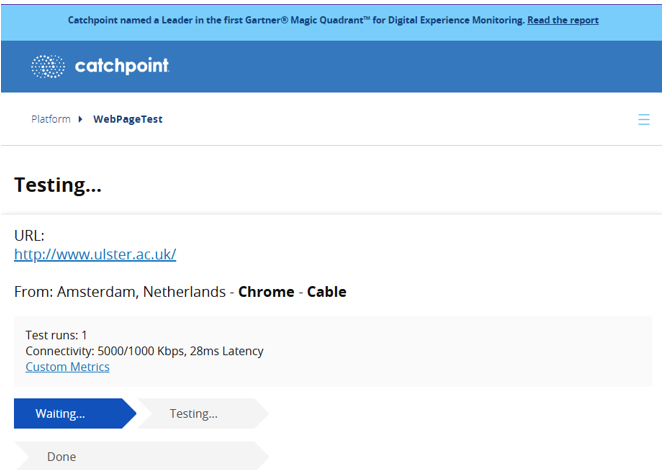
Изображение выглядит как текст, программное обеспечение, дисплей, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

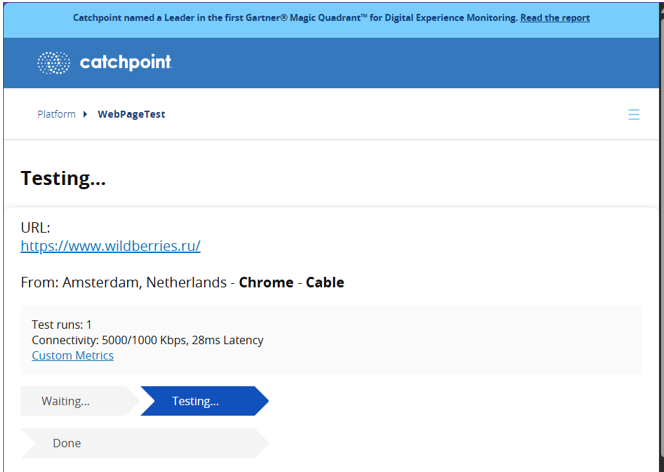
и счётчика пакетов HTTP

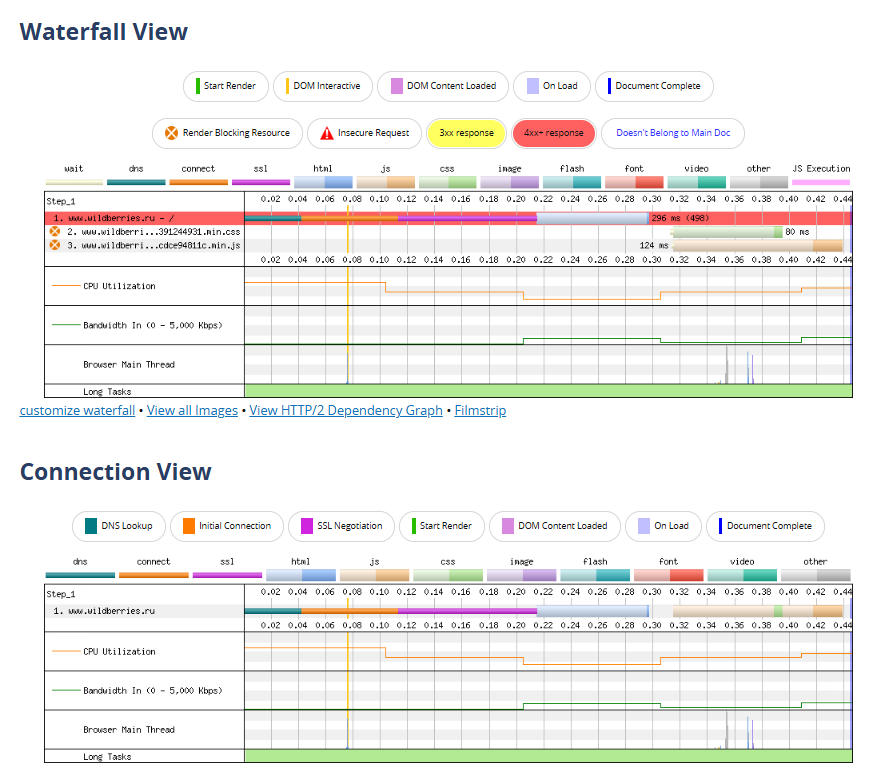


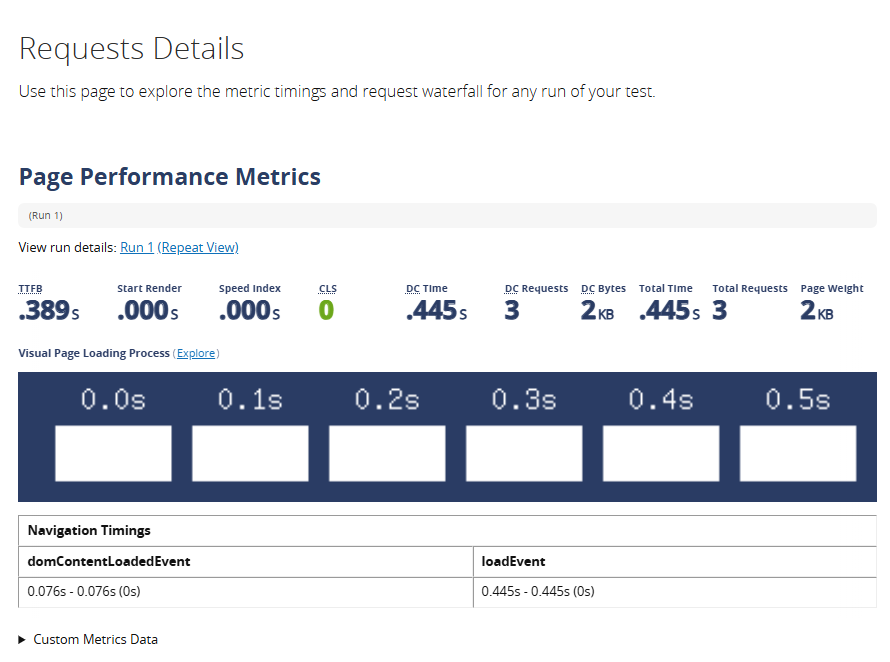
<https://www.webpagetest.org/.>



[www.ulster.ac.uk](http://www.ulster.ac.uk)







Задание 5. Отправка отчета на проверку

Изображение выглядит как текст, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Контрольные вопросы

Вот краткие и структурированные ответы на контрольные вопросы по протоколу HTTP и смежным темам:

Что такое протокол HTTP и для чего он используется?

HTTP — это протокол для передачи гипертекста и данных в интернете. Он используется для обмена информацией между клиентом (например, браузером) и сервером, чтобы загружать веб-страницы, отправлять данные форм, работать с API и т.д.

Каковы основные методы HTTP и какова их функция?

Основные методы: GET (получить данные), POST (отправить данные для обработки), PUT (полностью обновить ресурс), DELETE (удалить ресурс), PATCH (частично обновить ресурс), HEAD (получить только заголовки ответа), OPTIONS (узнать поддерживаемые методы).

Объясните разницу между методами GET и POST.

GET используется для получения данных, параметры передаются в URL, запрос кэшируется и безопасен. POST отправляет данные на сервер в теле запроса, не кэшируется, может менять состояние сервера (например, создание записи).

Что такое статус-коды HTTP и какова их роль в протоколе?

Статус-коды — это трёхзначные числа в ответе сервера, которые указывают результат обработки запроса. Они помогают клиенту понять, успешен ли запрос, требуется ли перенаправление или возникла ошибка.

Перечислите основные статус-коды HTTP и объясните их значение.

200 — OK (успешно), 201 — Created (ресурс создан), 301 — Moved Permanently (постоянное перенаправление), 302 — Found (временное перенаправление), 304 — Not Modified (контент не изменился), 400 — Bad Request (неверный запрос), 401 — Unauthorized (требуется аутентификация), 403 — Forbidden (доступ запрещён), 404 — Not Found (ресурс не найден), 500 — Internal Server Error (ошибка сервера).

Что такое заголовки HTTP и какую информацию они содержат?

Заголовки HTTP — это пары «ключ-значение» в запросе и ответе, которые передают метаданные. Они содержат информацию о клиенте, сервере, типе контента, кэшировании, куках, авторизации и т.д.

Какова роль заголовка User-Agent в HTTP-запросах?

User-Agent идентифицирует клиентское приложение (браузер, бот, мобильное приложение), отправляющее запрос. Сервер может использовать эту информацию для адаптации ответа (например, версии сайта для мобильных устройств).

Что такое HTTPS и как он отличается от HTTP?

HTTPS — это защищённая версия HTTP, которая использует шифрование TLS/SSL для обеспечения конфиденциальности и целостности данных. Отличие: HTTPS шифрует весь трафик между клиентом и сервером, предотвращая перехват данных.

Как работает механизм кэширования в HTTP?

Кэширование позволяет сохранять копии ресурсов (например, изображений, CSS) на стороне клиента или промежуточных прокси, чтобы сократить время загрузки и нагрузку на сервер. Управляется через заголовки (Cache-Control, Expires, ETag, Last-Modified).

Что такое сессия в контексте HTTP и как она реализуется?

Сессия — это состояние пользователя на сайте между несколькими запросами. Поскольку HTTP не сохраняет состояние, сессии реализуются через куки (с уникальным идентификатором), которые связывают запросы с данными на сервере.

Как использовать Wireshark для захвата HTTP-трафика?

Запустите Wireshark, выберите сетевой интерфейс (например, Wi-Fi или Ethernet), начните захват и отфильтруйте трафик по протоколу HTTP (фильтр http) или порту 80/443.

Что такое фильтры в Wireshark и как они помогают в анализе трафика?

Фильтры позволяют отображать только определённые пакеты по заданным критериям (например, http.request, ip.src==192.168.1.1, tcp.port==443). Это помогает изолировать нужный трафик и упрощает анализ.

Как можно просмотреть содержимое HTTP-запроса и ответа в Wireshark?

Выберите пакет HTTP, в деталях пакета найдите раздел «Hypertext Transfer Protocol». Разверните его, чтобы увидеть заголовки. Для просмотра тела запроса/ответа используйте опцию «Follow TCP Stream» или «Follow HTTP Stream».

Что такое TCP и как он взаимодействует с HTTP?

TCP — это транспортный протокол, обеспечивающий надёжную, упорядоченную доставку данных. HTTP работает поверх TCP: сначала устанавливается TCP-соединение (через трёхстороннее рукопожатие), затем по нему передаются HTTP-запросы и ответы.

Каковы особенности работы HTTP/2 по сравнению с HTTP/1.1?

HTTP/2 использует бинарный формат (вместо текстового), мультиплексирование (несколько запросов в одном соединении), приоритизацию, сжатие заголовков и Server Push (отправка ресурсов до запроса клиента). Это повышает скорость и эффективность.

Что такое HTTP-заголовок Referer и для чего он используется?

Referer (с опечаткой в спецификации) указывает URL предыдущей страницы, с которой пользователь перешёл на текущую. Используется для аналитики, логирования и иногда для контроля доступа (например, защита от горячих ссылок на изображения).

Каковы основные принципы работы с куками в HTTP?

Куки — это небольшие данные, которые сервер отправляет в заголовке Set-Cookie, а клиент сохраняет и возвращает в заголовке Cookie при последующих запросах. Используются для аутентификации, сессий, персонализации.

Что такое Content-Type и как он влияет на обработку данных?

Content-Type — это заголовок, указывающий тип данных в теле запроса или ответа (например, text/html, application/json). Он помогает клиенту и серверу правильно интерпретировать и обрабатывать контент (рендерить HTML, парсить JSON и т.д.).

Как можно использовать Wireshark для анализа проблем с производительностью HTTP?

В Wireshark можно анализировать задержки между запросами и ответами, время установки TCP-соединения, повторные передачи, размеры пакетов, использовать фильтры для медленных запросов (например, http.time > 0.5), а также статистику по протоколам.

Что такое REST и как он связан с протоколом HTTP?

REST — это архитектурный стиль для построения веб-сервисов, который использует HTTP как протокол передачи. Он опирается на методы HTTP (GET, POST, PUT, DELETE) для операций с ресурсами, представленными в виде URI, и обычно использует JSON/XML для обмена данными.