

## Лабораторная работа № 10

Тема: "Знакомство с анализаторами трафика. Wireshark."

Цель: Ознакомить студентов с использованием программы Wireshark для анализа сетевого трафика и основными принципами работы анализаторов трафика.

Необходимое оборудование/программное обеспечение:

- компьютеры с установленной операционной системой Windows или Linux
- программа Wireshark.

Анализ сетевого трафика имеет важное значение в современных сетевых средах по нескольким причинам:

**Безопасность сети:** Анализ сетевого трафика позволяет обнаруживать и предотвращать сетевые атаки и угрозы безопасности. Путем изучения сетевого трафика можно обнаружить аномальную активность, сканирование портов, попытки вторжения, вредоносные программы и другие подозрительные действия. Это помогает оперативно реагировать на угрозы, принимать меры по усилению безопасности и защищать сеть от несанкционированного доступа и потенциальных атак.

**Диагностика и устранение неполадок:** Анализ сетевого трафика помогает в идентификации и устранении проблем, связанных с работой сети. Путем изучения трафика можно выявить проблемы с пропускной способностью, задержками, потерей пакетов, неправильной конфигурацией устройств и другими факторами, которые могут вызывать неполадки в сети. Анализ трафика позволяет оперативно определить и исправить проблемы, минимизируя время простоя и обеспечивая более эффективное функционирование сети.

**Оптимизация сетевых ресурсов:** Анализ трафика помогает оптимизировать использование сетевых ресурсов. Посредством изучения распределения трафика, типов использованных протоколов и нагрузки на сеть можно определить узкие места, неэффективное использование ресурсов и возможности для оптимизации. Это позволяет принимать меры по улучшению производительности сети, балансировке нагрузки, оптимизации пропускной способности и улучшению качества обслуживания пользователей.

**Мониторинг и управление сетью:** Анализ сетевого трафика позволяет осуществлять мониторинг и управление сетью. Изучение трафика позволяет отслеживать активность пользователей, контролировать использование сетевых ресурсов, анализировать производительность и эффективность сети. Это помогает администраторам сети принимать информированные решения, планировать масштабирование сети и улучшать ее работу в соответствии с требованиями бизнеса и пользователями.

В целом, анализ сетевого трафика является важным инструментом для обеспечения безопасности, диагностики проблем, оптимизации ресурсов и эффективного управления сетью. Этот процесс позволяет получить глубокое понимание работы сети, выявить уязвимости и проблемы, а также принять меры для их устранения и улучшения работы сети в целом.

### Программы – анализаторы трафика (сниферы)

Некоторые из известных программ-сниферов (анализаторов трафика), которые широко используются для захвата и анализа сетевого трафика:

1. Wireshark: Одна из самых популярных и мощных программ для анализа сетевого трафика. Поддерживает множество протоколов и обладает широкими возможностями для изучения и фильтрации трафика.

2. **tcpdump**: Утилита командной строки для захвата и анализа сетевого трафика в Unix-подобных операционных системах. Предоставляет простой и гибкий интерфейс для фильтрации трафика и записи результатов в файл.
3. **tshark**: Консольная версия Wireshark, которая позволяет захватывать и анализировать сетевой трафик из командной строки. Она предоставляет мощные функции, аналогичные Wireshark, и может быть использована для автоматизации анализа трафика.
4. **Microsoft Network Monitor**: Программа от Microsoft для захвата и анализа сетевого трафика в операционных системах Windows. Обладает интуитивным интерфейсом и поддерживает различные протоколы.
5. **Capsa Network Analyzer**: Коммерческое программное обеспечение для анализа сетевого трафика, предоставляющее широкий спектр функций, включая захват, анализ, мониторинг и отчетность.
6. **Etercap**: Снифер и инструмент анализа сети с открытым исходным кодом. Предоставляет возможности для перехвата и анализа пакетов, осуществления атак типа "человек посередине" (Man-in-the-Middle) и других функций.
7. **Cain & Abel**: Комплексный инструмент для тестирования безопасности сетей, который включает в себя функции снифинга трафика, взлома паролей, анализа протоколов и других возможностей.
8. **Colasoft Capsa**: Программное обеспечение для анализа сетевого трафика с широкими возможностями мониторинга и анализа, включая захват и анализ пакетов, мониторинг пропускной способности и анализ производительности сети.

### **Wireshark.**

Wireshark - это мощная и популярная программа для анализа сетевого трафика. Она предоставляет обширный набор инструментов для захвата, анализа и отображения пакетов данных, передаваемых по сети. Вот некоторые из основных возможностей Wireshark:

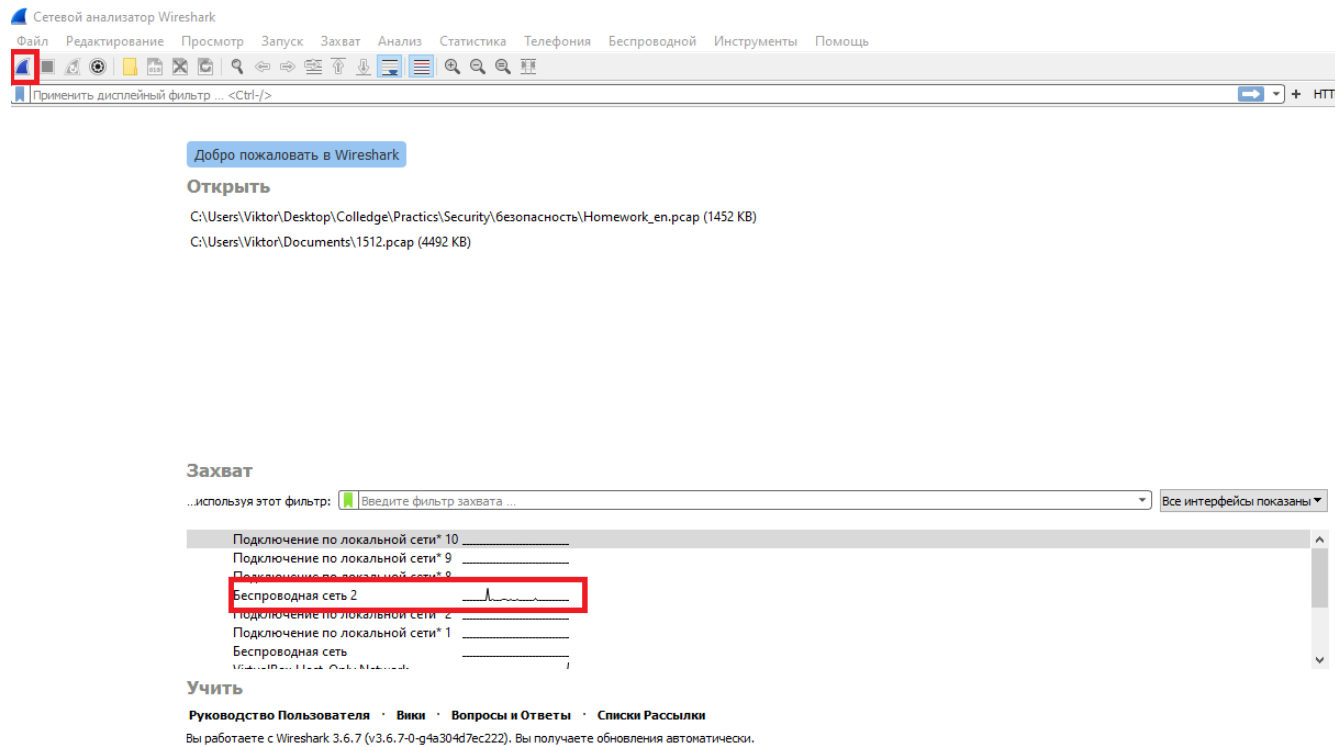
1. **Захват пакетов**: Wireshark позволяет захватывать пакеты данных, передаваемые по сети, на основе выбранного сетевого интерфейса. Он поддерживает различные типы интерфейсов, включая Ethernet, Wi-Fi, USB и другие. В результате захвата пакетов Wireshark создает подробный список пакетов для последующего анализа.
2. **Анализ протоколов**: Wireshark распознает и анализирует множество сетевых протоколов, включая TCP, UDP, IP, HTTP, DNS, FTP, SSH, SSL и многие другие. Он декодирует и отображает информацию, содержащуюся в пакетах, позволяя анализировать потоки данных, заголовки протоколов, поля и значения.
3. **Фильтрация трафика**: Wireshark имеет мощные средства фильтрации, позволяющие выбирать и отображать только нужные пакеты и информацию. Фильтры могут быть применены на основе адресов источника и назначения, протоколов, типов пакетов, содержимого полей и других параметров, что облегчает анализ и сокращает объем отображаемой информации.
4. **Подсветка и цветовая кодировка**: Wireshark позволяет пользователю настраивать цветовую кодировку пакетов в соответствии с выбранными правилами и фильтрами. Это упрощает визуальное отслеживание и выделение определенных типов пакетов или событий, таких как ошибки, предупреждения или атаки.
5. **Статистика и отчетность**: Wireshark предоставляет разнообразные статистические данные о захваченном трафике, включая количество пакетов, объем переданных данных, пропускную способность, распределение протоколов и другие параметры. Он также позволяет создавать отчеты на основе этих данных для дальнейшего анализа или представления другим пользователям.
6. **Поддержка платформ**: Wireshark доступен для различных операционных систем, включая Windows, macOS и Linux, что делает его универсальным инструментом для анализа сетевого трафика в различных средах.

Wireshark обладает еще большим количеством функций и возможностей, которые позволяют анализировать и исследовать сетевой трафик на глубоком уровне. Он широко

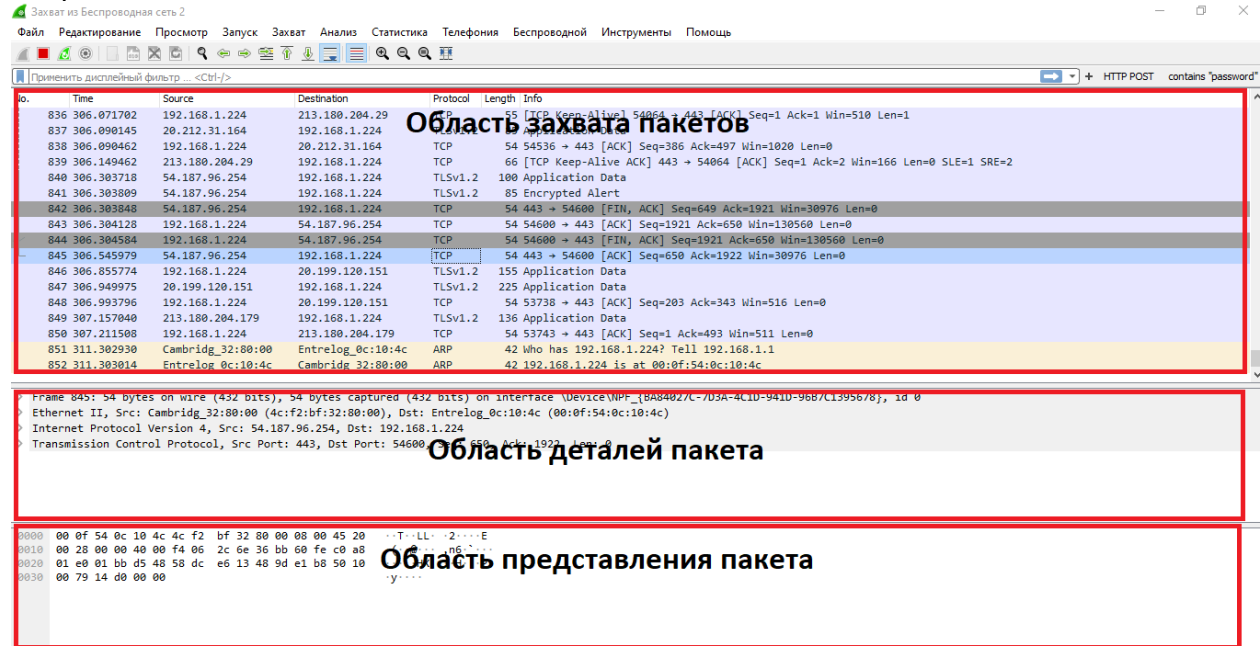
применяется специалистами в области сетевой безопасности, системного администрирования, разработки и отладки сетевых приложений, а также в образовательных целях.

Работа в Wireshark

1. Запустить программу. Откроется стартовое окно, где нужно выбрать интересующий сетевой интерфейс. Кроме названия интерфейса, видно и его активность. Что бы начать захват трафика, нужно либо дважды кликнуть на нужном интерфейсе, либо нажать кнопку захвата:



Откроется окно захвата пакетов:



Окно захвата в программе Wireshark состоит из трех основных областей:

Область захвата пакетов: Эта область находится в верхней части окна и представляет собой список захваченных пакетов данных. Каждая строка в списке представляет один

пакет с информацией, такой как номер пакета, время захвата, источник и назначение, протоколы и другие сведения. Эта область дает общую информацию о захваченных пакетах и может быть использована для выбора конкретных пакетов для дальнейшего анализа.

**Область деталей пакета:** Расположена под областью захвата пакетов. Здесь отображаются подробные сведения о выбранном пакете, включая заголовки протоколов, поля данных и другую информацию. Wireshark предоставляет декодирование данных, чтобы пользователь мог легко прочитать и понять содержимое пакета. Эта область позволяет анализировать каждый пакет на более глубоком уровне.

**Область представления пакета:** Расположена ниже области деталей пакета. Здесь отображается содержимое выбранного пакета в различных представлениях, таких как бинарный, шестнадцатеричный, текстовый и другие форматы. Пользователь может выбрать представление, которое наиболее удобно для анализа содержимого пакета. Это позволяет просматривать данные пакета в различных форматах и облегчает изучение и понимание информации.

Вместе эти три области предоставляют разнообразные инструменты и информацию для захвата, анализа и отображения сетевого трафика в Wireshark. Они позволяют пользователю изучать пакеты данных на разных уровнях детализации, анализировать протоколы, фильтровать трафик и извлекать полезную информацию для решения различных задач, связанных с сетевым анализом.

### **Фильтрация трафика.**

Wireshark предоставляет мощные возможности фильтрации для упрощения анализа трафика. Фильтры позволяют выбирать и отображать только нужные пакеты и информацию, исключая ненужные данные. Вот некоторые из наиболее распространенных фильтров отображения, которые могут быть использованы в Wireshark:

1. **Фильтры по адресу:** Вы можете фильтровать пакеты на основе адреса источника или назначения. Например, вы можете использовать фильтр `ip.src == 192.168.0.1` для отображения только пакетов, исходящих от указанного IP-адреса.
2. **Фильтры по протоколу:** Вы можете фильтровать пакеты на основе протокола. Например, фильтр `http` отобразит только пакеты, относящиеся к протоколу HTTP.
3. **Фильтры по содержимому:** Вы можете фильтровать пакеты на основе содержимого полей или данных. Например, фильтр `tcp.port == 80` отобразит только пакеты с TCP-портом 80 (обычно используемым для HTTP).
4. **Фильтры по типу пакета:** Вы можете фильтровать пакеты на основе их типа или состояния. Например, фильтр `tcp.flags.syn == 1` отобразит только пакеты с установленным флагом SYN в TCP.
5. **Фильтры по времени:** Вы можете фильтровать пакеты на основе времени захвата. Например, вы можете использовать фильтр `frame.time >= "2021-01-01 00:00:00"` для отображения пакетов, захваченных после указанной даты и времени.
6. **Комбинированные фильтры:** Вы также можете комбинировать несколько фильтров для более точной фильтрации. Например, вы можете использовать фильтр `ip.src == 192.168.0.1 && tcp.port == 80` для отображения только HTTP-пакетов, исходящих от указанного IP-адреса.

Wireshark также предлагает расширенный синтаксис фильтров, который позволяет создавать более сложные выражения и комбинировать различные условия. Вы также можете сохранять и применять фильтры для повторного использования.

Пример использования фильтра:

Wireshark 2.10.0 (64-bit) - Беспроводная сеть 2

Файл Редактирование Просмотр Запуск Захват Анализ Статистика Телефония Беспроводной Инструменты Помощь

http

No.	Time	Source	Destination	Protocol	Length	Info
32337	1117.875926	52.209.66.100	192.168.1.224	HTTP	550	HTTP/1.1 200 OK
32373	1124.755150	192.168.1.224	52.209.66.100	HTTP	712	POST /cloudquery.php HTTP/1.1
32375	1124.875311	52.209.66.100	192.168.1.224	HTTP	550	HTTP/1.1 200 OK
32388	1125.194784	192.168.1.224	52.209.66.100	HTTP	920	POST /cloudquery.php HTTP/1.1
32390	1125.313705	52.209.66.100	192.168.1.224	HTTP	908	HTTP/1.1 200 OK
32437	1135.431465	192.168.1.224	52.209.64.157	HTTP	482	POST /scan HTTP/1.1 (application/x-www-form-urlencoded)
32439	1135.549790	52.209.64.157	192.168.1.224	HTTP	322	HTTP/1.1 200 OK
32593	1164.777619	192.168.1.224	52.209.66.100	HTTP	712	POST /cloudquery.php HTTP/1.1
32596	1164.892670	52.209.66.100	192.168.1.224	HTTP	550	HTTP/1.1 200 OK
32607	1165.772412	192.168.1.224	52.209.66.100	HTTP	712	POST /cloudquery.php HTTP/1.1
32609	1165.891323	52.209.66.100	192.168.1.224	HTTP	550	HTTP/1.1 200 OK
32620	1166.335071	192.168.1.224	52.209.66.100	HTTP	720	POST /cloudquery.php HTTP/1.1
32622	1166.453782	52.209.66.100	192.168.1.224	HTTP	550	HTTP/1.1 200 OK
32634	1166.847291	192.168.1.224	52.209.66.100	HTTP	712	POST /cloudquery.php HTTP/1.1
32636	1166.966954	52.209.66.100	192.168.1.224	HTTP	550	HTTP/1.1 200 OK
32655	1170.045025	192.168.1.224	52.209.66.100	HTTP	712	POST /cloudquery.php HTTP/1.1
32658	1170.167176	52.209.66.100	192.168.1.224	HTTP	550	HTTP/1.1 200 OK

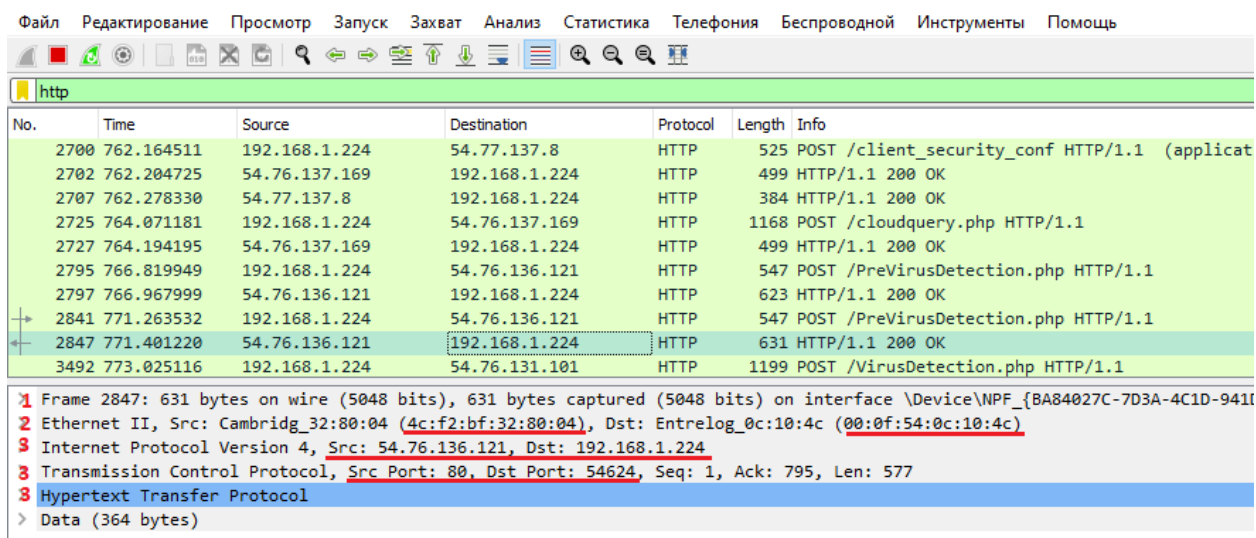
Ниже приведен список некоторых из самых популярных фильтров Wireshark, которые широко используются для анализа сетевого трафика:

1. Фильтр по IP-адресу источника: `ip.src == <IP-адрес>`. Например, `ip.src == 192.168.0.1` отобразит только пакеты, исходящие от указанного IP-адреса.
2. Фильтр по IP-адресу назначения: `ip.dst == <IP-адрес>`. Например, `ip.dst == 192.168.0.1` отобразит только пакеты, адресованные указанному IP-адресу.
3. Фильтр по протоколу: `proto == <протокол>`. Например, `proto == http` отобразит только пакеты, относящиеся к протоколу HTTP.
4. Фильтр по порту источника: `tcp.srcport == <порт>`. Например, `tcp.srcport == 80` отобразит только пакеты с TCP-портом 80.
5. Фильтр по порту назначения: `tcp.dstport == <порт>`. Например, `tcp.dstport == 443` отобразит только пакеты с TCP-портом 443.
6. Фильтр по содержимому данных: `data contains "<строка>"`. Например, `data contains "password"` отобразит только пакеты, содержащие строку "password" в поле данных.
7. Фильтр по типу пакета: `http, tcp, udp, icmp` и т. д. Можно использовать для отображения только пакетов, относящихся к указанному типу.
8. Фильтр по размеру пакета: `frame.len == <размер>`. Например, `frame.len > 1000` отобразит только пакеты размером больше 1000 байт.
9. Фильтр по времени захвата: `frame.time >= "<дата и время>"`. Например, `frame.time >= "2021-01-01 00:00:00"` отобразит только пакеты, захваченные после указанной даты и времени.
10. Фильтр по методу используемому в протоколе HTTP. `http.request.method==POST`
11. Комбинированные фильтры: Можно комбинировать несколько условий с помощью операторов логического И (`&&`) и логического ИЛИ (`||`). Например, `ip.src == 192.168.0.1 && tcp.dstport == 80` отобразит только пакеты, исходящие от указанного IP-адреса и имеющие TCP-порт 80.

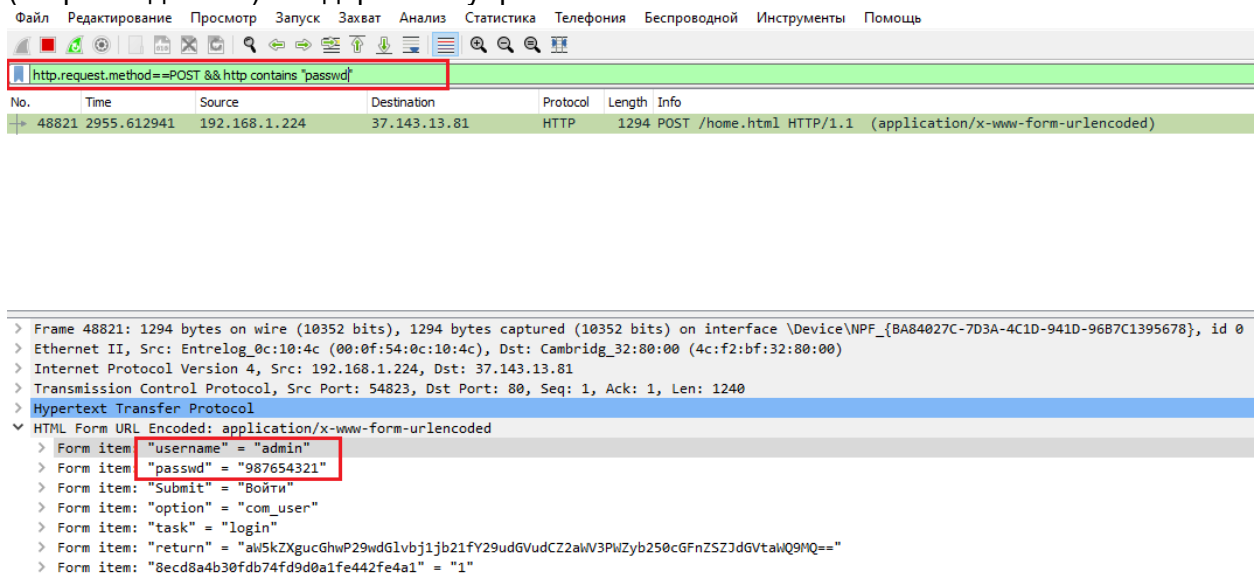
Область деталей пакета в Wireshark представляет собой панель, которая отображает подробную информацию о выбранном пакете. Здесь вы можете просмотреть различные аспекты пакета, включая заголовки протоколов, поля, значения и другую сопутствующую информацию. Вот некоторые основные элементы и функции области деталей пакета:

1. Общая информация: В верхней части области деталей пакета отображается общая информация о пакете, включая номер пакета, время захвата, длительность источника и назначения, а также размер пакета.
2. Информация о фрейме: Здесь вы найдете информацию о физическом уровне (фрейме), такую как тип Ethernet-фрейма, MAC-адреса и другие связанные атрибуты.
3. Заголовки протоколов: В области деталей пакета отображаются заголовки различных протоколов, присутствующих в пакете. Вы можете раскрыть каждый заголовок, чтобы просмотреть его поля и значения.

4. Поля протоколов: Каждый заголовок протокола разделен на отдельные поля с соответствующими значениями. Вы можете щелкнуть на поле, чтобы просмотреть его подробности, включая описание, значение, формат и другую информацию.
5. Дерево пакета: Слева от области деталей пакета находится дерево пакета, которое отображает иерархическую структуру пакета. Вы можете развернуть каждый уровень дерева, чтобы просмотреть подробности и поля на каждом уровне протокола.
6. Поле "Значение": В нижней части области деталей пакета отображается окно "Значение", где вы можете просмотреть содержимое выбранного поля протокола в шестнадцатеричном, десятичном или текстовом формате.



В примере пакет полученный в результате применения фильтра по HTTP методу POST (отправка данных) и содержимому "passwd":



HTTP – протокол без шифрования, поэтому данные передаются в открытом виде и могут быть перехвачены.

### Анализ всей сессии.

Wireshark позволяет просматривать и анализировать полный контент сетевой сессии между двумя узлами. Это полезный инструмент для изучения взаимодействия и обмена данными между клиентом и сервером. Вот как использовать эту функцию:

Выберите пакет, относящийся к сетевой сессии, которую вы хотите проанализировать. Например, это может быть пакет соединения TCP или запрос HTTP.

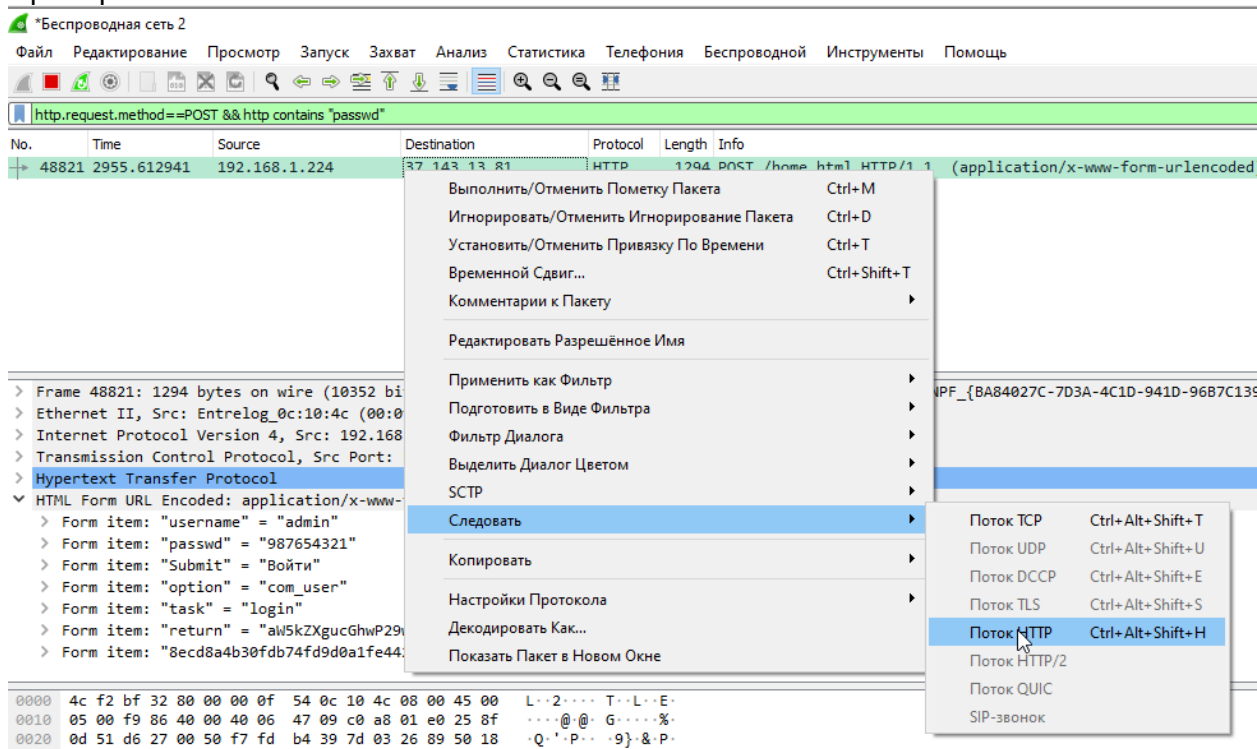
Щелкните правой кнопкой мыши на выбранном пакете, а затем выберите в контекстном меню пункт "Follow" (Анализ-следовать). В подменю выберите соответствующий протокол, например, "TCP Stream" или "HTTP Stream".

Откроется новое окно, отображающее весь контент сетевой сессии между выбранными узлами. В этом окне вы увидите полный поток данных, включая отправленные и полученные пакеты, заголовки протоколов, тело сообщений и другую информацию.

В окне Анализ-следовать вы можете просматривать данные в текстовом или шестнадцатеричном формате. Вы можете переключаться между различными представлениями, используя вкладки или опции в верхней части окна.

В некоторых случаях, например, при анализе HTTP-сессии, Wireshark может показывать дополнительную информацию, такую как разделение заголовков и содержимого, разбор частей запроса или ответа, и т. д.

Пример:





## Результат:

Wireshark · Следовать HTTP Поток (tcp.stream eq 196) · Беспроводная сеть 2

```
POST /home.html HTTP/1.1
Host: www.unkniga.ru
Connection: keep-alive
Content-Length: 215
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://www.unkniga.ru
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.771
YaBrowser/23.11.2.771 Yowser/2.5 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://www.unkniga.ru/index.php?option=com_content&view=frontpage&Itemid=1
Accept-Encoding: gzip, deflate
Accept-Language: ru,en;q=0.9,ba;q=0.8
Cookie: unkniga2_tpl=unkniga2; _ym_uid=1701965187212649130; _ym_d=1701965187; _ga=GA1.2.1285629297.1701965187; unkniga_tpl=unkniga; PHPSESSID=806vg050gjldifq9fkip7m7900; ff09d1b99fed20bc4d39eadb62d94653=nvt20o2ft2jb0ah71ph8nrmn15; _ym_isad=1; _gid=GA1.2.282704883.1702639506; _ga_D8F52SBJK3=GS1.2.1702695333.3.0.1702695333.0.0.0

username=admin&passwd=987654321&Submit=%D0%92%D0%BE%D0%B9%D1%82%D0%B8&option=com_user&task=login&return=aW5kZXgucGhwP29wdG1vbWJ1b21fY29udGVudCZ2aWV3PWZyb250cGFnZSZJdGVtYWQ9MQ%3D%3D&ecd8a4b30fdb74fd9d0a1fe442fe4a1=1HTTP/1.1 200 OK
Server: nginx/1.3.14
Date: Sat, 16 Dec 2023 03:04:27 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 155
Connection: keep-alive
X-Powered-By: PHP/5.3.21
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"

<html><head><meta http-equiv="refresh" content="0;http://www.unkniga.ru/index.php?option=com_content&view=frontpage&Itemid=1" /></head><body></body></html>GET /index.php?option=com_content&view=frontpage&Itemid=1 HTTP/1.1
Host: www.unkniga.ru
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

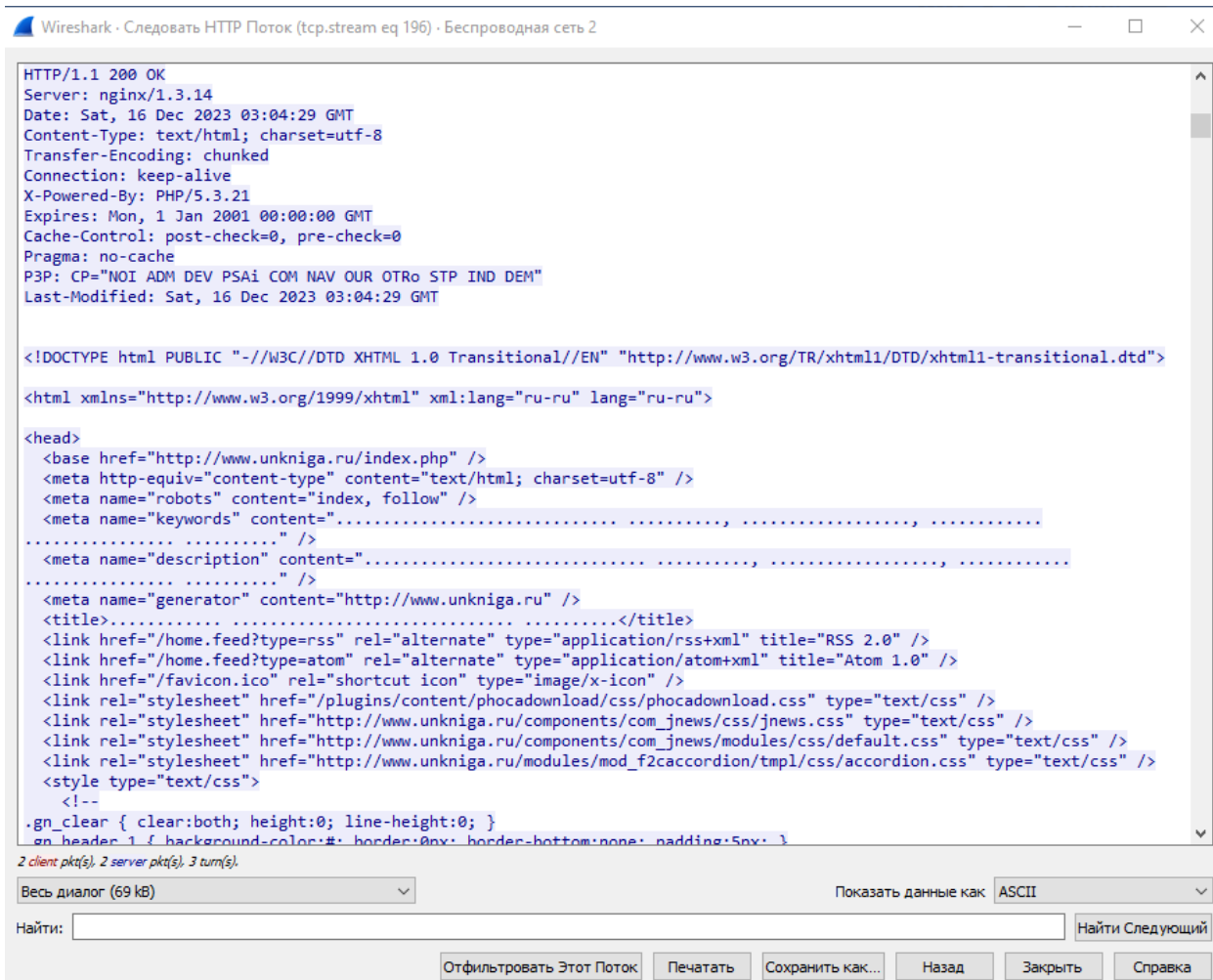
Пакет 48841. 2 client pkt(s), 2 server pkt(s), 3 turn(s). Щёлкните, чтобы выбрать.

Весь диалог (69 kB)

Показать данные как ASCII

Найти:





## Statistics -> Conversations

Функция "Статистика-диалоги" (Statistics -> Conversations) в Wireshark позволяет анализировать статистическую информацию о сетевых диалогах, происходящих между узлами в захваченном трафике. Она предоставляет обзорную информацию о количестве и характеристиках коммуникации между различными узлами в сети. Вот как использовать функцию Статистика-диалоги:

1. Откройте файл захвата или начните захват трафика в Wireshark.
2. Выберите меню "Статистика" (Statistics) в главном меню Wireshark, а затем выберите пункт "Диалоги" (Conversations).
3. Откроется окно "Статистика-диалоги", где вы увидите список всех диалогов между узлами в захваченном трафике. Каждая строка представляет собой отдельный диалог с указанием источника и назначения.
4. В окне "Статистика-диалоги" вы можете видеть различные статистические данные о каждом диалоге, такие как количество пакетов, байт, продолжительность, используемые протоколы и другую связанную информацию.
5. Вы можете сортировать диалоги по различным столбцам, щелкнув на заголовок столбца. Например, вы можете отсортировать диалоги по количеству пакетов или объему переданных данных. Если отсортировать по порту назначения, можно понять какие протоколы использованы в диалогах (сессиях).
6. Щелкнув правой кнопкой мыши на диалоге, вы можете выбрать дополнительные опции, такие как просмотр деталей диалога или фильтрация пакетов, связанных с этим диалогом.

Функция "Статистика-диалоги" полезна при анализе сетевого трафика, позволяя визуализировать и суммировать общую активность между узлами. Это может быть полезно при определении наиболее активных участников сети, идентификации проблемных соединений или анализе потребления ресурсов.

Wireshark · Conversations · Беспроводная сеть 2

Ethernet · 12		IPv4 · 132		IPv6 · 5		TCP · 245		UDP · 611					
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.1.224	54753	52.209.66.100	80	12	2109	6	1277	6	1277	832 165.54398	0.4629	22 k	14 k
192.168.1.224	54754	52.209.66.100	80	12	2117	6	1285	6	1285	832 166.08373	0.4850	21 k	13 k
192.168.1.224	54755	52.209.66.100	80	12	2109	6	1277	6	1277	832 166.61506	0.4669	21 k	14 k
192.168.1.224	54756	52.209.66.100	80	12	2109	6	1277	6	1277	832 169.79101	0.4896	20 k	13 k
192.168.1.224	54763	52.209.64.157	80	10	1404	5	860	5	860	544 319.35240	0.3626	18 k	12 k
192.168.1.224	54771	54.76.137.169	80	12	1906	6	1197	6	1197	709 707.82118	0.4533	21 k	12 k
192.168.1.224	54797	54.76.137.169	80	10	1404	5	860	5	860	544 219.42645	0.3425	20 k	12 k
192.168.1.224	54804	37.143.13.81	80	90	72 k	40	3156	50	69 k	2419.66067	554.8349	45	999
192.168.1.224	54805	37.143.13.81	80	24	1343	21	1157	3	186 k	2419.66140	554.8341	16	2
192.168.1.224	54823	37.143.13.81	80	87	74 k	34	4046	53	70 k	2955.55163	564.4985	57	996
192.168.1.224	54824	37.143.13.81	80	18	1019	15	833	3	186 k	2955.55227	555.2301	12	2
192.168.1.224	54826	54.76.6.106	80	10	1837	5	904	5	933 k	1994.76317	0.3426	21 k	21 k
192.168.1.224	54827	54.255.136.181	80	10	1043	5	550	5	493 k	1995.02807	0.3321	13 k	11 k
192.168.1.224	54837	54.194.213.128	80	10	1458	5	910	5	548 k	1119.51801	0.3352	21 k	13 k
192.168.1.224	54854	54.76.137.169	80	12	1962	6	1181	6	781 k	450.07898	0.4586	20 k	13 k
192.168.1.224	54855	52.209.64.157	80	10	1412	5	868	5	544 k	450.23489	0.3430	20 k	12 k
192.168.1.224	54856	52.209.64.157	80	10	1404	5	860	5	544 k	450.46888	0.3432	20 k	12 k
192.168.1.224	54857	52.209.64.157	80	10	1348	5	804	5	544 k	450.70090	0.3472	18 k	12 k
192.168.1.224	54858	52.209.64.157	80	10	1644	5	1100	5	544 k	450.93848	0.3584	24 k	12 k
192.168.1.224	54859	52.209.64.157	80	10	1452	5	908	5	544 k	451.18727	0.3700	19 k	11 k
192.168.1.224	54860	52.209.64.157	80	10	1364	5	820	5	544 k	451.44373	0.3616	18 k	12 k
192.168.1.224	54861	52.209.64.157	80	10	1356	5	812	5	544 k	451.69725	0.3427	18 k	12 k
192.168.1.224	54862	52.209.64.157	80	10	1356	5	812	5	544 k	451.92908	0.3484	18 k	12 k

☐ Разрешение имен   
 ☐ Ограничить соответственно дисплейному фильтру   
 ☐ Абсолютное время старта   
 Conversation Типы ▾

Копировать ▾   
 Отслеживание Потока...   
 График...   
 **Заккрыть**   
 Справка

## Экспорт объектов.

Функция "Файл -> Экспортировать объекты -> HTTP" (File -> Export Objects -> HTTP) в Wireshark позволяет экспортировать HTTP-объекты, такие как изображения, HTML-страницы, аудио- и видеофайлы, из захваченного сетевого трафика. Это полезная функция при анализе веб-сайтов, загрузке медиафайлов и изучении содержимого HTTP-трафика. Вот как использовать эту функцию:

Откройте файл захвата или начните захват трафика в Wireshark.

Выберите меню "Файл" (File) в главном меню Wireshark, затем выберите пункт "Экспортировать объекты" (Export Objects) и затем "HTTP".

Откроется окно "Экспортировать объекты HTTP", которое отображает список всех HTTP-объектов, найденных в захваченном трафике.

В окне "Экспортировать объекты HTTP" вы можете просмотреть список объектов, включая URL, размер файла и другую информацию.

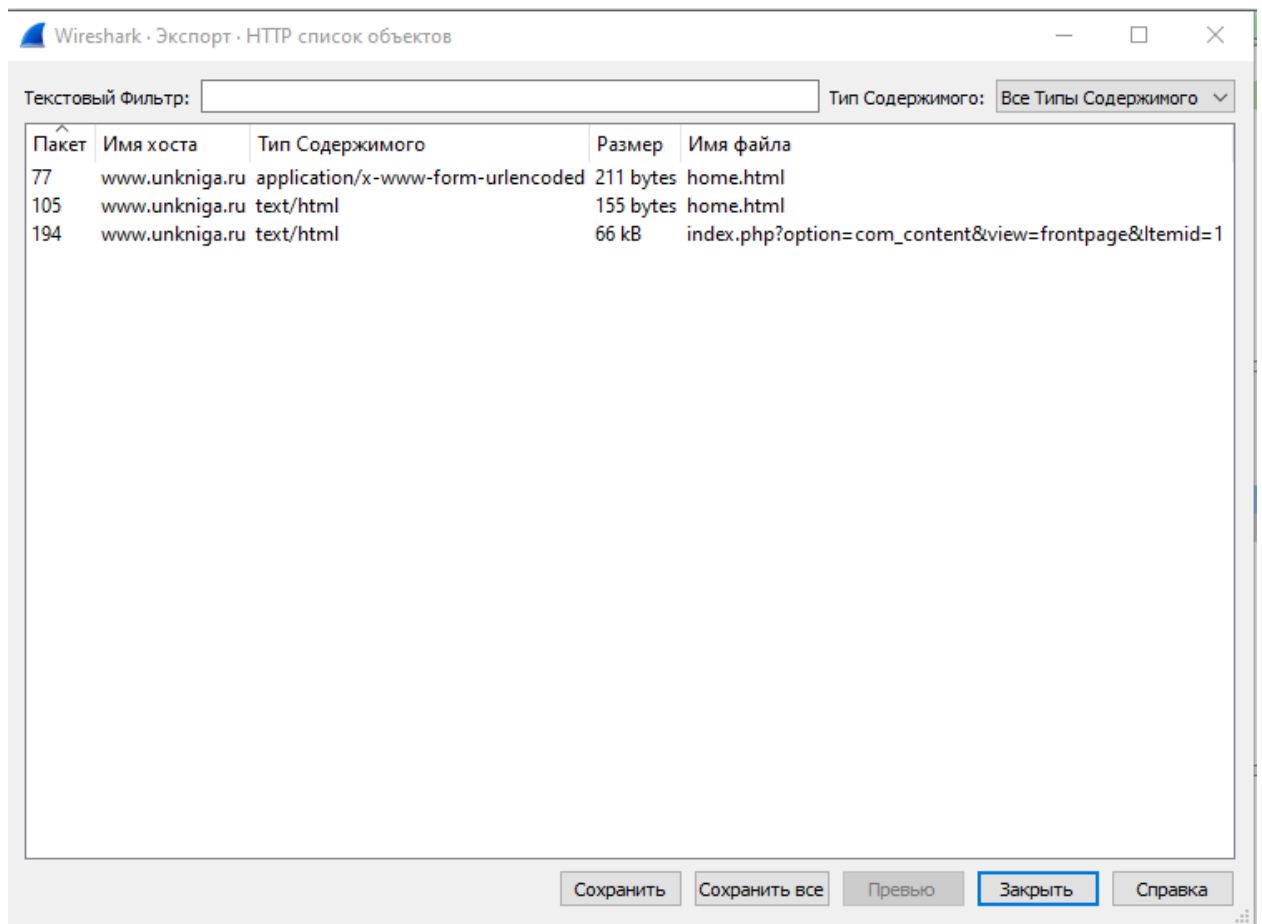
Выберите объекты, которые вы хотите экспортировать. Вы можете выбрать один или несколько объектов, щелкнув на них.

Щелкните на кнопке "Сохранить" (Save), чтобы указать местоположение, куда будут экспортированы выбранные объекты. Выберите папку и имя файла для сохранения.

Нажмите кнопку "Сохранить" (Save) в диалоговом окне сохранения файлов, чтобы начать экспорт выбранных HTTP-объектов.

Wireshark сохранит выбранные HTTP-объекты в указанной вами папке. В зависимости от типа объекта, они могут быть сохранены в виде отдельных файлов или одного архива.

Экспортирование объектов HTTP полезно при исследовании веб-трафика, изучении веб-страниц, загрузке медиафайлов или извлечении других ресурсов, передаваемых по протоколу HTTP. Это позволяет более детально анализировать содержимое HTTP-сообщений и извлекать интересные вас данные для дальнейшего исследования.



Задание:

1. Захватите трафик во время выполнения ARP-запросов и ответов. Изучите структуру и содержимое ARP-пакетов. Определите процесс разрешения MAC-адресов на IP-адреса и обратно.

2. Захватите трафик во время процесса назначения IP-адреса DHCP-клиенту. Изучите структуру и содержимое DHCP-пакетов. Определите различные типы сообщений DHCP, такие как DHCP Discover, DHCP Offer, DHCP Request и т. д. Проанализируйте процесс назначения IP-адреса.

3. Захватите трафик между клиентом и сервером во время выполнения TCP-соединения. Изучите процесс установления, поддержки и завершения соединения TCP. Определите флаги TCP, такие как SYN, ACK, FIN и т. д., и их использование в различных этапах соединения.

4. Захватите трафик во время просмотра веб-страницы в браузере. Исследуйте HTTP-запросы и ответы, чтобы понять структуру и содержимое протокола. Определите заголовки запросов и ответов, а также параметры, передаваемые в URL и тела сообщений. Сравните различные методы HTTP (GET, POST, PUT и т. д.) и их использование в разных сценариях.

5. Захватите трафик во время выполнения DNS-запросов, например, попробуйте открыть веб-сайт в браузере. Когда DNS-запрос выполнен остановите запись трафика и изучите структуру и содержимое DNS-пакетов и проанализируйте следующую информацию:

- Query Name: Имя домена, запрошенного у DNS-сервера.
- Query Type: Тип DNS-запроса (например, A, AAAA, MX).

- Response: Ответ DNS-сервера, включая IP-адреса или другие данные.

Отчет должен содержать:

- скриншоты Wireshark, подтверждающие выполнение заданий.
- выводы.