

ПРАКТИЧЕСКАЯ РАБОТА № 1

Изучение средств мониторинга и анализа сетевого трафика.

ЦЕЛЬ РАБОТЫ

1. Знать принципы анализа сетевого трафика.
2. Научиться использовать сетевой анализатор (сниффер Wireshark).
3. Научиться анализировать сетевой трафик на примере основных сетевых протоколов. протоколов ARP, IP и ICMP.

АНАЛИЗ СЕТЕВОГО ТРАФИКА И ПАКЕТОВ С ИСПОЛЬЗОВАНИЕМ СНИФФЕРА «WIRESHARK»

Sniffer (от англ. to sniff – нюхать) – это сетевой анализатор трафика, программа или программно-аппаратное устройство, предназначенное для перехвата и последующего анализа, либо только анализа сетевого трафика, предназначенного для других узлов.

Перехват трафика может осуществляться:

- обычным «прослушиванием» сетевого интерфейса (метод эффективен при использовании в сегменте концентраторов (хабов) вместо коммутаторов (свичей), в противном случае метод малоэффективен, поскольку на сниффер попадают лишь отдельные фреймы);
- подключением сниффера в разрыв канала;
- ответвлением (программным или аппаратным) трафика и направлением его копии на сниффер;
- через анализ побочных электромагнитных излучений и восстановление таким образом прослушиваемого трафика;
- через атаку на канальном (2-й) или сетевом (3-й) уровне, приводящую к перенаправлению трафика жертвы или всего трафика сегмента на сниффер с последующим возвращением трафика в надлежащий адрес.

В начале 1990-х широко применялся хакерами для захвата пользовательских логинов и паролей. Широкое распространение хабов позволяло захватывать трафик без больших усилий в больших сегментах сети.

Снифферы применяются как в благих, так и в деструктивных целях. Анализ прошедшего через сниффер трафика, позволяет:

- Отслеживать сетевую активность приложений.
- Отлаживать протоколы сетевых приложений.
- Локализовать неисправность или ошибку конфигурации.
- Обнаружить паразитный, вирусный и закольцованный трафик, наличие которого увеличивает нагрузку сетевого оборудования и каналов связи.

- Выявить в сети вредоносное и несанкционированное ПО, например, сетевые сканеры, флудеры, троянские программы, клиенты пиринговых сетей и другие.
- Перехватить любой незашифрованный (а порой и зашифрованный) пользовательский трафик с целью узнавания паролей и другой информации.

Постепенно из инструментов, предназначенных только для диагностики, снифферы постепенно превратились в средства для исследований и обучения. Например, они постоянно используются для изучения динамики и взаимодействий в сетях. В частности, они позволяют легко и наглядно изучать тонкости сетевых протоколов. Наблюдая за данными, которые посылает протокол, вы можете глубже понять его функционирование на практике, а заодно увидеть, когда некоторая конкретная реализация работает не в соответствии со спецификацией.

На сегодняшний момент существует достаточно большое количество хороших реализаций снифферов. Некоторое из них:

- Tcpdump (<http://www.tcpdump.org/>) – консольный вариант сниффера. Портитован почти подо все наиболее распространенные ОС;
- Wireshark (<http://www.wireshark.org/>) ранее известен под названием Ethereal;
- WinDump <http://www.winpcap.org/windump>; и др.
- Tshark – консольный вариант сниффера от разработчиков Wireshark

СНИФФЕР WIRESHARK

Программа Wireshark является одной из самых удобных реализаций sniffеров. Портитована на большое количество платформ. Распространяется абсолютно бесплатно и доступна по ссылке <https://www.wireshark.org/#download>

Базовый принцип работы sniffеров

На рисунке 1 схематично изображена структура сетевой подсистемы ОС. Вся базовая инфраструктура реализована в виде драйверов и работает в режиме ядра. Пользовательские процессы и реализации прикладных протоколов, в частности интерфейс sniffера работают в пользовательском режиме.

На рисунке отображены 2 пользовательских процесса («сетевой процесс 1» и «сетевой процесс 2»).

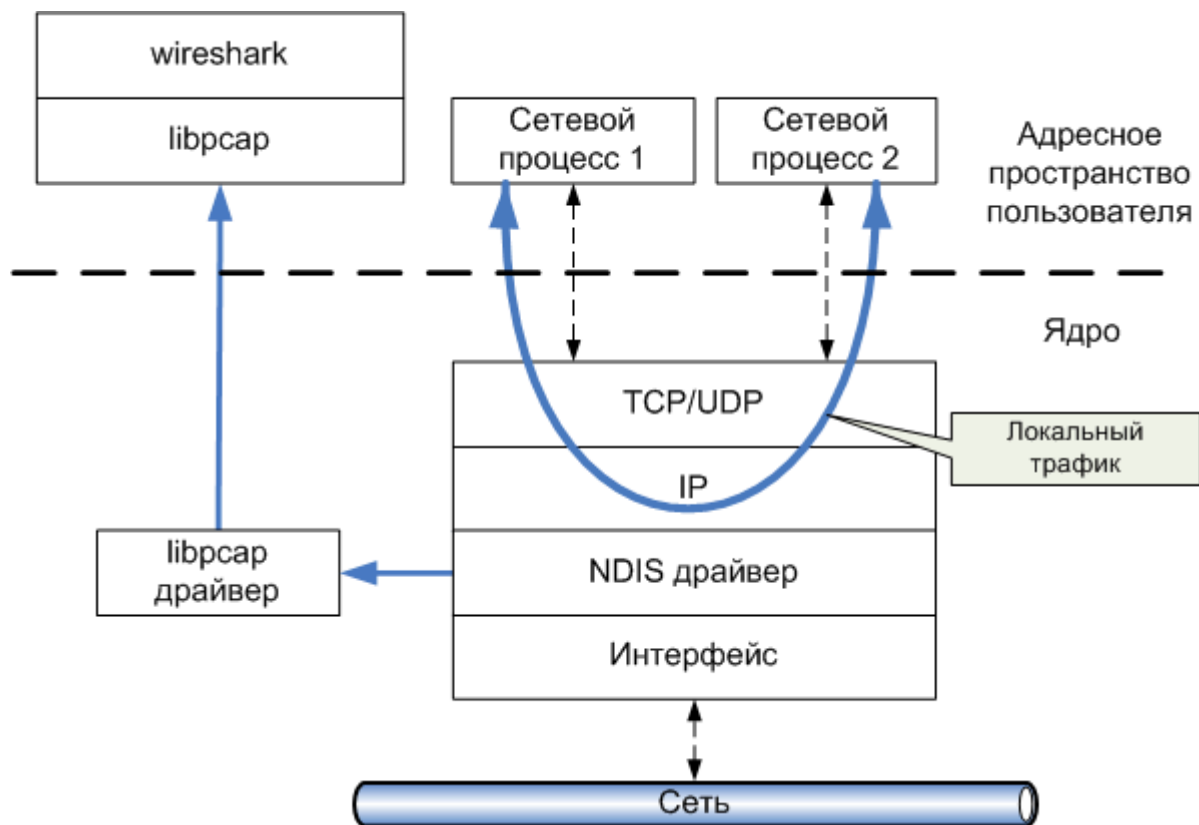


Рисунок 1 – Принцип захвата sniffером сетевого трафика

Сниффер получает данные из сетевого интерфейса не через стандартный стек протоколов TCP/IP (который используют большинство сетевых приложений)

Основными компонентами sniffера являются: драйвер для захвата пакетов (**libpcap драйвер**), интерфейсная библиотека (**libpcap**) и интерфейс пользователя (**Wireshark**). Библиотека **libpcap** (реализация под ОС Windows носит название

WinPcap - <http://www.winpcap.org>) – универсальная сетевая библиотека, самостоятельно реализующая большое количество сетевых протоколов и

работающая непосредственно с NDIS (Network Driver Interface Specification) драйверами сетевых устройств. На базе данной библиотеки реализовано большое количество сетевых программ, в частности сниффер Wireshark.

Сниффер использует библиотеку в режиме «захвата» пакетов, т.е. может получать копию VCEX данных проходящих через драйвер сетевого интерфейса. Изменения в сами данные не вносятся!

Стоит отметить, что снифферы вносят дополнительную нагрузку на процессор, т.к. могут обрабатывать достаточно объемный сетевой трафик, в особенности для высокоскоростных соединений (Fast Ethernet, Gigabit Ethernet и др.).

Использование программы Wireshark

Wireshark захватывать пакеты из сети, и анализировать их структуру. Программа также позволяет анализировать структуру пакетов из файла, содержащего трафик, полученными другими анализаторами пакетов, например, программой «tcpdump» (unix/linux).

При запуске программы отображаются сетевые интерфейсы из которых можно выбрать активный интерфейс для анализа. (рисунок 2).

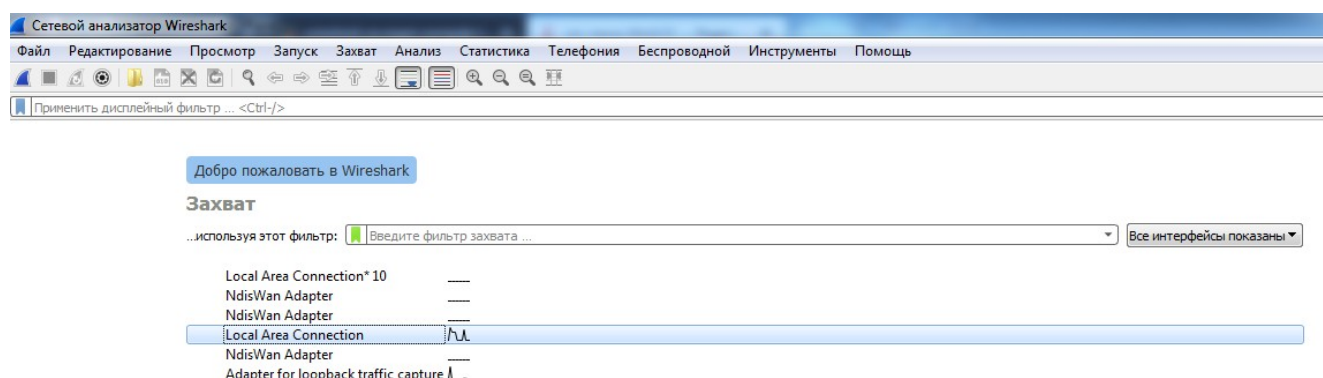


Рисунок 2 – Выбор интерфейса анализа

После этого откроется основное окно программы Wireshark. В стандартном режиме окно сниффера делится на 3 основные панели: список захваченных пакетов, «анализатор» протоколов и исходные данные пакетов. Размер каждого фрейма можно менять по своему усмотрению.

Рассмотрим эти панели подробнее.

Меню, панель инструментов
Дисплейный Фильтр

Список захваченных пакетов

Информационное поле с детальной информацией по выбранному пакету

Содержимое пакета в 16-чной и текстовой формах

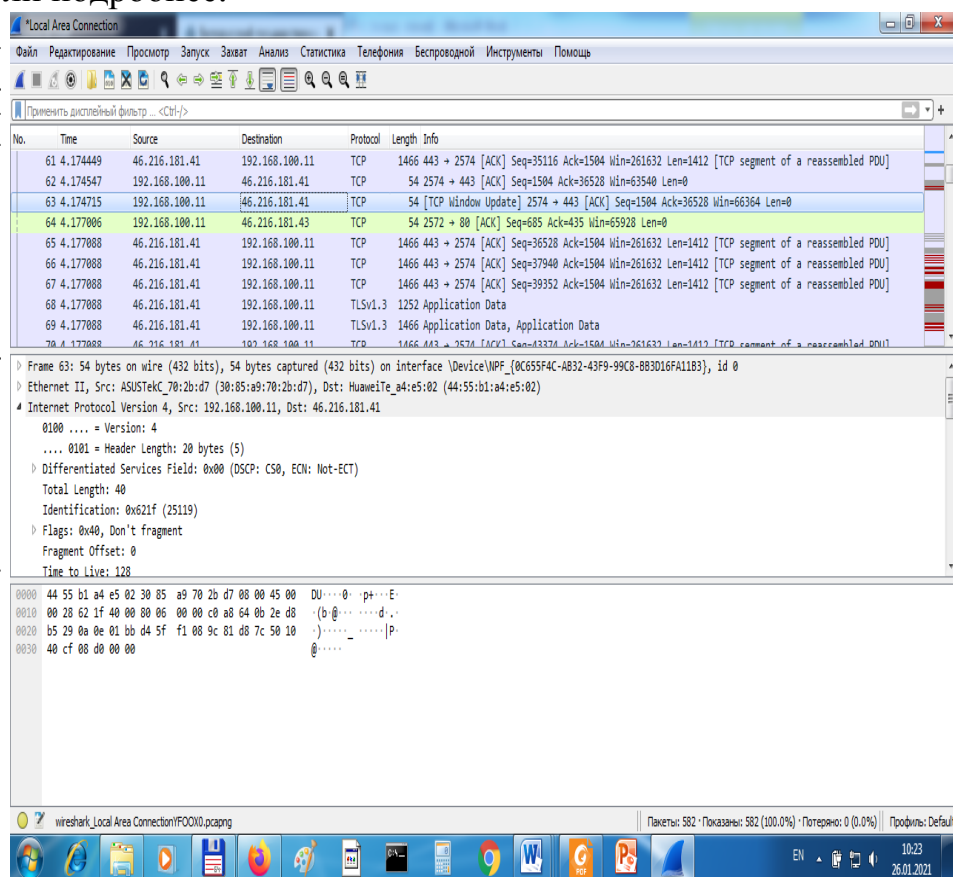


Рисунок 2 – Внешний вид программы

Верхняя панель содержит список пакетов, захваченных из сети. Список можно отсортировать по любому полю (в прямом или обратном порядке) – для этого нажать на заголовок соответствующего поля.

Каждая строка содержит следующие поля (по умолчанию):

- порядковый номер пакета (No.);
- время поступления пакета (Time);
- источник пакета (Source);
- пункт назначения (Destination);
- протокол (Protocol);
- информационное поле (Info).

Список отображаемых полей настраивается в Edit/Preferences/Columns. Для того, чтобы изменения возымели эффект необходимо перезапустить программу, предварительно нажав кнопку Save.


При нажатии правой кнопки мыши на том или ином пакете, появится контекстное меню. Нажатием на среднюю кнопку мыши можно пометить группу интересующих пакетов.



Средняя панель содержит т.н. «дерево протоколов» для выбранного в верхнем окне пакета. В этой панели в иерархическом виде для выбранного в верхнем окне захваченного пакета отображается вложенность протоколов в соответствии с моделью взаимодействия открытых систем OSI. По нажатию на правую кнопку мыши вызывается контекстное меню. При «раскрытии» каждого

из протокола нажатием на значек «+» слева, выводятся поля данных соответствующих протоколов.

Нижняя панель содержит шестнадцатеричное представление выбранного пакета. При выборе того или иного поля в средней панели автоматически будет подсвечиваться соответствующий участок 16-ого представления.

ЗАХВАТ ПАКЕТОВ

Для начала захвата пакетов необходимо задать параметры захвата. В частности, указать сетевой интерфейс, с которого и будет осуществляться захват пакетов. Это действие доступно так же через меню «Захват→Опции» (рисунок 3) или кнопкой на панели инструментов . В качестве дополнительных параметров захвата можно указать следующие – фильтр захвата (будет рассмотрен далее). По нажатию на соответствующую кнопку можно применить тот или иной фильтр отбора (из ранее сохраненных). Если таковых не имеется, его можно указать явно в строке редактирования.

«Для начала мониторинга сетевой активности нужно нажать «Start» или кнопкой . Остановить захват пакетов можно кнопкой «Стоп» .

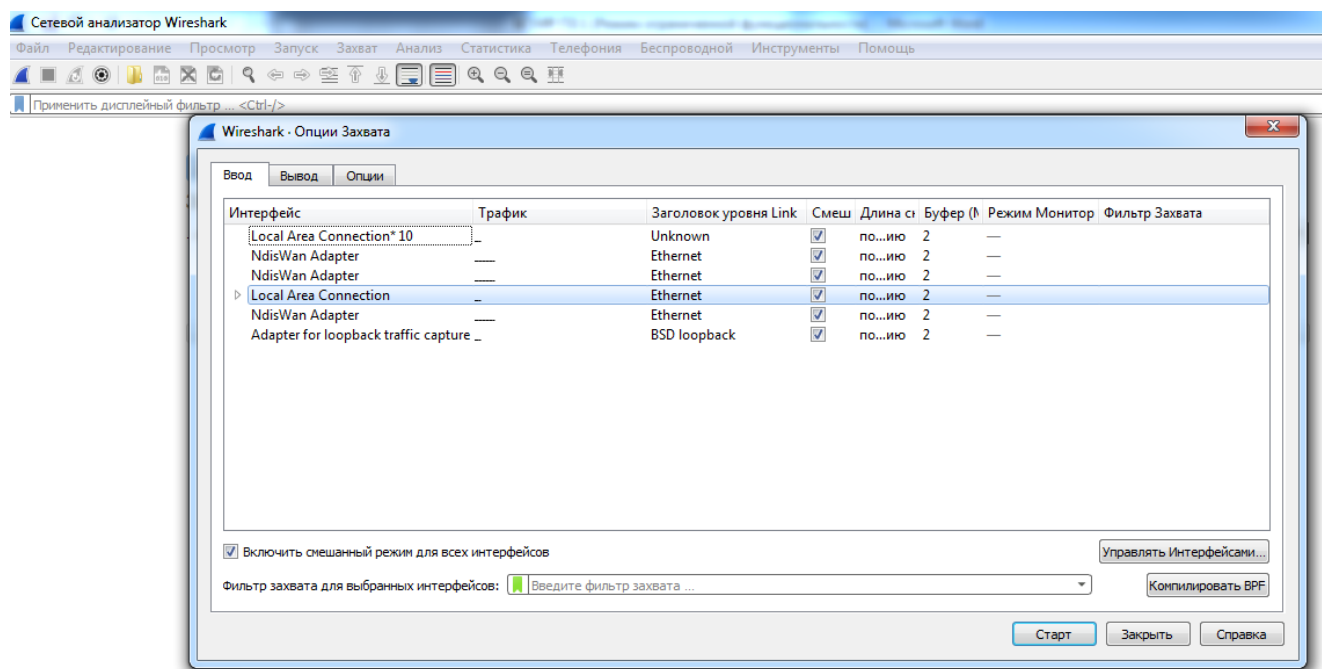


Рисунок 3 Опции захвата пакетов

ФИЛЬТРАЦИЯ ПАКЕТОВ

Если запустить сниффер без дополнительных настроек, он будет «захватывать» все пакеты, проходящие через сетевые интерфейсы (см. рис. 1). Вообще, для общего ознакомления с процессами, происходящими в сети, очень полезно пронаблюдать активность сетевых протоколов в реальных условиях работы системы в сети. Пронаблюдать все разнообразие протоколов, запросов, ответов и др. событий.

При целенаправленном использовании сниффера очень часто необходимо выборочно отображать или захватывать пакеты по некоторым заданным критериям. Для этих целей служат фильтры отображения и захвата, соответственно.

Типы фильтрации трафика

Существует два варианта фильтрации пакетов: на этапе захвата и на этапе отображения пользователю. В первом случае эффективность работы сниффера и потребляемые им системные ресурсы значительно ниже, нежели во втором случае. Это объясняется тем, что при достаточно интенсивном сетевом трафике и продолжительном времени захвата все пакеты должны быть захвачены и сохранены либо в память, либо на дисковое устройство. Самые простые подсчеты могу показать, что даже для 100-мегабитной сети системных ресурсов хватит на непродолжительное время. Фильтрация захвата уже на момент получения пакета гораздо эффективнее, однако в таком случае она должна быть реализована на уровне самих драйверов захвата. Данный факт, естественно, усложняет реализацию сниффера. Wireshark поддерживает оба варианта фильтрации.

Фильтры отображения

Фильтры отображения представляют собой достаточно мощное средство отображения трафика. Фильтры задаются в строке, располагающейся вверху основного экрана («Filter:»). Простейший фильтр отображения, позволяет отобрать пакеты по тому или иному протоколу. Для этого в строке требуется указать название протокола (например TCP). После этого в верхнем окне останутся пакеты, принадлежащие этому протоколу.

Для работы с фильтрами можно вызвать окно «**Analyze/Display Filters**». Можно сохранять созданные выражения под определенными именами для последующего использования и т.д.

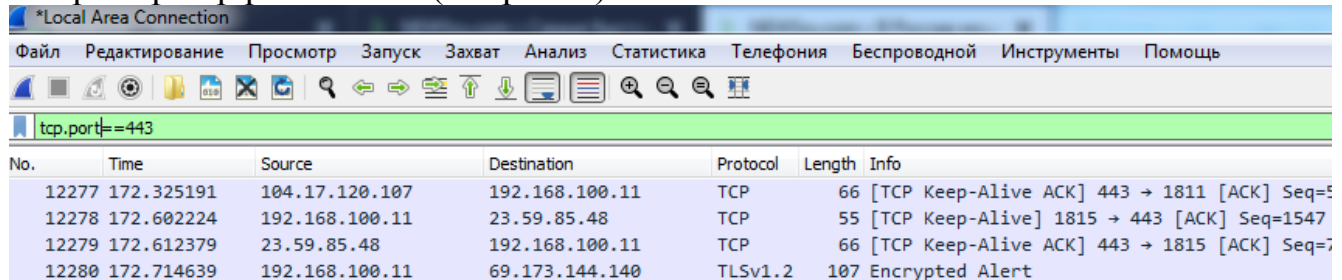
С помощью логических операций (синтаксис языка Си) можно составлять логические выражения. Логическая истина - 1, ложь - 0.

Список поддерживаемых логических операций:

eq	==	равенство
ne	!=	не равно
gt	>	больше чем
Lt	<	меньше чем
ge	>=	больше равно

Le <= меньше равно

Например: tcp.port == 443 (см. рис. 4).



No.	Time	Source	Destination	Protocol	Length	Info
12277	172.325191	104.17.120.107	192.168.100.11	TCP	66	[TCP Keep-Alive ACK] 443 → 1811 [ACK] Seq=5
12278	172.602224	192.168.100.11	23.59.85.48	TCP	55	[TCP Keep-Alive] 1815 → 443 [ACK] Seq=1547
12279	172.612379	23.59.85.48	192.168.100.11	TCP	66	[TCP Keep-Alive ACK] 443 → 1815 [ACK] Seq=7
12280	172.714639	192.168.100.11	69.173.144.140	TLSv1.2	107	Encrypted Alert

Рисунок 4

Мастер построения фильтров отображения доступен через кнопку «Выражение Дисплейного фильтра» в меню «Анализ» (см. рис. 5).

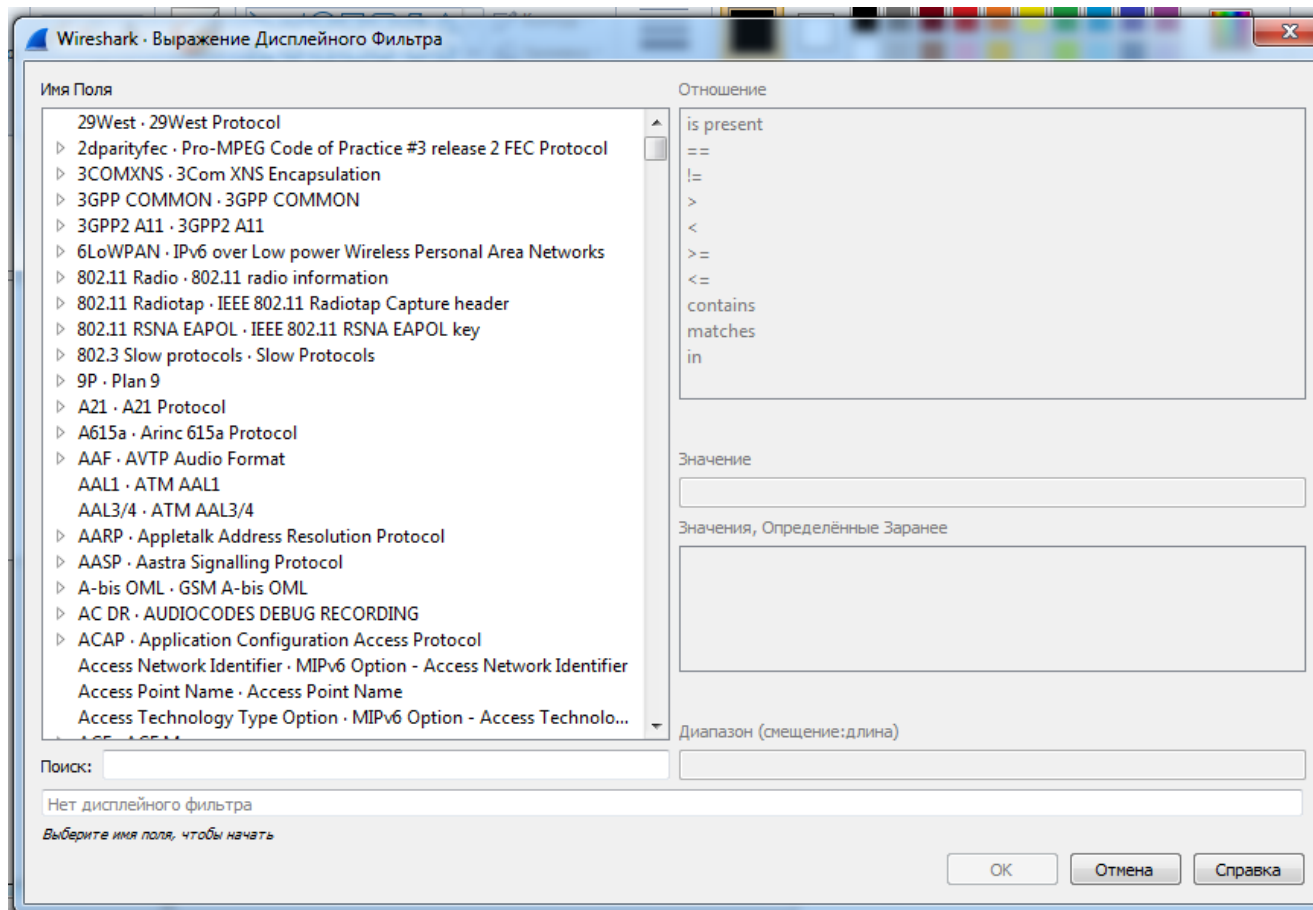


Рисунок 5 – Мастер построения дисплейного фильтра

Фильтры захвата

С помощью данных фильтров можно захватывать из сети только те пакеты, которые подходят под критерий отбора. Если не задано никакого фильтра (по умолчанию), то будут захватываться все пакеты. В противном случае только

пакеты, для которых указанное выражение будет истинным. Синтаксис фильтров захвата несколько отличается от синтаксиса фильтров отображения. Выражение состоит из одного или более примитивов разделенных пробельными символами. На рис. 6 приведен пример фильтра для захвата пакетов, адресованных на 80-й порт (http) узла с ip-адресом 192.168.1.100.

Существует три различных типа примитивов: *type*, *dir*, *proto*.

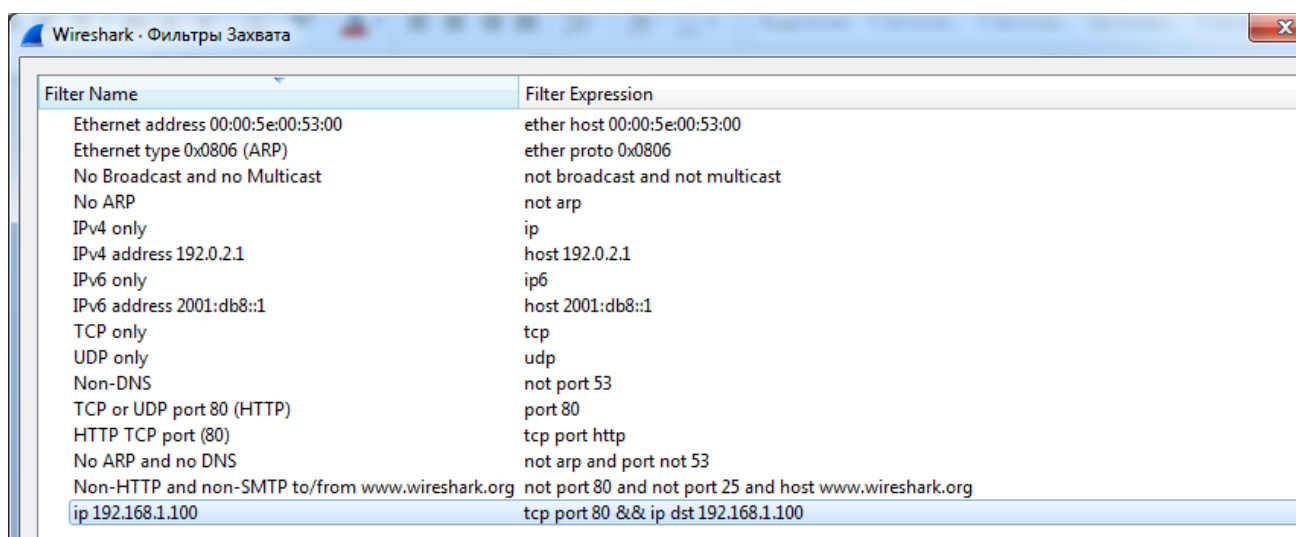
Спецификатор *type* определяет тип параметра. Возможные параметры: **host**; **net**; **port**.

Например:

- host linux
- net 192.168.128
- port 80

Если не указано никакого типа предполагается что это параметр **host**.

Спецификатор *dir* определяют направление передачи. Возможные направления: **src**; **dst**; **src or dst**; **src and dst**.



Filter Name	Filter Expression
Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	ether proto 0x0806
No Broadcast and no Multicast	not broadcast and not multicast
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	host 192.0.2.1
IPv6 only	ip6
IPv6 address 2001:db8::1	host 2001:db8::1
TCP only	tcp
UDP only	udp
Non-DNS	not port 53
TCP or UDP port 80 (HTTP)	port 80
HTTP TCP port (80)	tcp port http
No ARP and no DNS	not arp and port not 53
Non-HTTP and non-SMTP to/from www.wireshark.org	not port 80 and not port 25 and host www.wireshark.org
ip 192.168.1.100	tcp port 80 && ip dst 192.168.1.100

Рисунок 6 – Пример создания фильтра захвата

Например:

- dst net 192.168.128
- src or dst port http

Если не определено направление то предполагается направление «**src or dst**». Для протоколов типа point-to-point используются спецификаторы **inbound** и **outbound**.

Спецификатор *proto* определяют тип протокола, которому принадлежит пакет.

Возможные протоколы: **ether**; **fddi**; **tr**; **ip/ipv6**; **arp/rarp**; **decent**; **tcp**; **udp**.

Например:

- arp net 192.168.128
- tcp port 80

Если протокол не определен, то будут захватываться пакеты всех протоколов. То есть «net ctam» означает «(ip or arp or rarp) net ctam»; «port 53» означает «(tcp or udp) port 53».

Также существует несколько специальных спецификаторов, которые не попадают в описанные выше случаи:

- *gateway*;
- *broadcast*;
- *less*;
- *greater*;
- *арифметические выражения*.

Сложные фильтры захвата строятся с использованием логических выражений.

Список операций:

not	!	отрицание
and	&&	конкатенация (логическое И)
or		альтернатива (логическое ИЛИ)

Примеры фильтров захвата

Ниже рассмотрены некоторые примеры построения фильтров захвата.

- Захват всех пакетов на сетевом интерфейсе хоста 192.168.1.2:
host 192.168.1.2
- Захват трафика между хостом host1 И хостами host2 ИЛИ host3:
host host1 and (host2 or host3)
- Захват всех IP-пакетов между хостом host1 и каждым хостом за исключением hostX:
ip host host1 and not hostX
- Захват пакетов ни сгенерированных ни адресованных локальными хостами:
ip and not net localnet
- Захват IP-пакетов размером больше чем 576 байт, проходящих через шлюз snup:
gateway snup and ip[2:2] > 576
- Захват всех ICMP пакетов, за исключением пакетов ping:
icmp[0] != 8 and icmp[0] != 0

Сниффер Wireshark позволяет выполнять различную статистическую обработку полученных данных. Все доступные операции находятся в меню «Статистика».

Задания к лабораторной работе

Задание 1. HTTP: Basic HTTP GET/response

- 1) запустить *Wireshark*;
- 2) настроить фильтр (http);
- 3) запустить процесс захвата трафика;
- 4) Найти URL с протоколом http. Для этого можно выполнить поисковый запрос в Google: `inurl:http -inurl:https`

В URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции, либо выбрать URL по номеру результата поиска, совпадающего с номером в группе;

- 5) остановить захват трафика.

В списке захваченных пакетов найдите пару HTTP сообщений (запрос- ответ): GET сообщение и ответ сервера. В информационном поле разверните строку, содержащую HTTP, и отметьте указанную ниже информацию.

1. Версия HTTP.
2. Принимаемые браузером языки.
3. IP-адреса вашего компьютера и сервера.
4. Код состояния HTTP. Что он означает?
5. Длина тела сообщения. (Содержимое поля заголовка объекта `ContentLength` указывает длину тела сообщения в октетах (десятичное число),

Задание 2. DNS

nslookup — утилита, предоставляющая пользователю интерфейс командной строки для обращения к системе DNS (проще говоря, DNS-клиент). Позволяет задавать различные типы запросов и запрашивать произвольно указываемые сервера.

- 1) очистить кэш DNS с помощью *ipconfig* (в командной строке): `ipconfig /flushdns`
- 2) очистить кэш браузера (<https://zen.yandex.ru/media/mhelp/kak-ochistit-kesh-brauzera-chrome-firefox-i-dr-5dd4e57c5e28df4e59a38609>)
- 3) запустить *Wireshark*;
- 4) настроить фильтр: `ip.addr == ваш_IP_адрес`;
- 5) запустить процесс захвата трафика;
- 6) в командной строке: `nslookup host` например, `nslookup bsuir.by` в качестве узла (host) использовать URL, в котором присутствуют любые три буквы из фамилии студента в латинской транскрипции;
- 5) остановить захват трафика. Прим.: Всего *nslookup* отправил три DNS-запроса и получил три DNS- ответа. Для дальнейшего анализа использовать последние два пакета. (Первые два набора запросов/ответов не генерируются стандартными интернет-приложениями и специфичны для *nslookup*.)

Ответьте на следующие вопросы.

1. Найдите DNS-запрос и ответ. Поверх какого протокола транспортного уровня работает DNS?
2. Укажите порты источника/назначения для DNS-запроса и DNS- ответа.
3. На какой IP-адрес отправлен DNS-запрос? Совпадает ли этот адрес с адресом вашего DNS-сервера?
4. Укажите тип DNS-запроса.
5. Что содержится в поле «Answers» DNS-ответа?

Задание 3. ICMP

- 1) запустить *Wireshark*;
- 2) настроить фильтр (icmp);
- 3) запустить процесс захвата трафика;
- 4) в командной строке: `ping -n 10 конечный_узел` например, `ping -n 10 bsuir.by` в качестве конечного узла использовать URL, в котором присутствуют любые три буквы из фамилии студента в латинской транскрипции;
- 5) остановить захват трафика.

Ответьте на следующие вопросы.

1. Сколько всего пакетов захватила программа? Почему?
2. Какой IP-адрес вашего компьютера, адрес назначения? 3. Проанализируйте ping request, отправленный с вашего компьютера: укажите тип и код ICMP. Какие еще поля содержит ICMP пакет? Сколько байт занимают поля «Checksum», «Sequence number», «Identifier»?
4. Проанализируйте ping reply: укажите тип и код ICMP. Какие еще поля содержит ICMP пакет?
5. Сколько байт занимают поля «Checksum», «Sequence number», «Identifier»?

Задание 4. DHCP

- 1) в командной строке: `ipconfig /release` (IP-адрес станет 0.0.0.0)
- 2) запустить *Wireshark*;
- 3) настроить фильтр (bootp);
- 4) запустить процесс захвата трафика;
- 5) в командной строке: `ipconfig /renew` (получение нового IP-адреса) и еще раз: `ipconfig /release ipconfig /renew`
- 6) остановить захват трафика.

Ответьте на следующие вопросы.

1. Поверх какого протокола транспортного уровня работает DHCP?
2. Поясните последовательность обмена первыми четырьмя пакетами Discover/Offer/Request/ACK. Укажите для каждого пакета номера портов источника, назначения
3. Какими значениями отличаются пакеты DHCP Discover и DHCP Request?
4. Укажите значения поля «Transaction-ID» для всех пакетов (Discover/Offer/Request/ACK), что это поле означает?
5. Укажите IP-адреса источника, назначения для всех пакетов.

Задание 5 Ethernet

- 1) очистить кэш браузера;
- 2) запустить *Wireshark*;
- 3) запустить процесс захвата трафика;
- 4) Найти URL с протоколом http. Для этого можно выполнить поисковый запрос в Google: `inurl:http -inurl:https`

В URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции, либо выбрать URL по номеру результата поиска, совпадающего с номером в группе

5) остановить захват трафика;

6) в меню «Анализ» → «Enabled Protocols» можно снять галочку IP: тогда в списке пакетов не будет отображаться информация по протоколам верхнего уровня (после IP) — (необязательный пункт).

Выберите кадр Ethernet, содержащий сообщение HTTP GET.

Ответьте на следующие вопросы.

1. Укажите 48-битный Ethernet адрес вашего компьютера.
2. Укажите 48-битный Ethernet адрес назначения.
3. Что это за адрес? (Адрес сервера?)
4. Укажите 16-чное значение двухбайтового поля «Type»: какому протоколу верхнего уровня оно соответствует?

Выберите кадр Ethernet, содержащий ответ HTTP. Ответьте на следующие вопросы.

1. Укажите значение Ethernet адреса источника. Какое устройство имеет такой адрес?
2. Укажите Ethernet адрес назначения: это адрес вашего компьютера?
3. Укажите 16-чное значение двухбайтового поля «Type»: какому протоколу верхнего уровня оно соответствует?

Задание 6 ARP

- 1) Запустить Wireshark.
- 2) очистить ARP кэш: `arp -d`. Проверить `arp`- таблицу командой: `arp -a` и убедиться, что записей нет.
- 3) запустить процесс захвата трафика;
- 4) с командной строки пропинговать шлюз
- 5) остановить захват трафика;
- 6) включить фильтр просмотра `arp`
- 7) Найти `arp`-запрос и `arp`-ответ

Ответьте на следующие вопросы.

1. Укажите 16-чные значения адресов источника и назначения в пакете, содержащем ARP запрос (ARP ответ).
2. Укажите 16-чное значение двухбайтового поля «Type»: какому протоколу верхнего уровня оно соответствует (для ARP запроса/ARP ответа)?
3. Укажите значение поля «opcode» (для ARP запроса/ARP ответа).
4. К какому уровню модели OSI можно отнести протокол ARP

Задание 7. FTP

- 1) запустить *Wireshark*;
- 2) запустить процесс захвата трафика;

- 3) скачать файл с FTP-сервера; в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;
- 4) остановить захват трафика;
- 5) настроить фильтр (ftp || ftp-data).

Ответьте на следующие вопросы.

1. Сколько байт данных содержится в пакете FTP-DATA? 2. Укажите IP-адреса FTP-сервера и вашего компьютера.
4. Укажите протокол транспортного уровня, который использует протокол FTP.
5. Укажите порт, который используется при передаче данных по протоколу FTP.

- Задание 8. UDP** 1) запустить *Wireshark*; 2) настроить фильтр (udp); 3) запустить процесс захвата трафика;
- 4) использовать команду nslookup (см. DNS);
 - 5) остановить захват трафика.

Ответьте на следующие вопросы.

1. Выберите один UDP пакет из списка пакетов. Сколько полей в UDP заголовке? Что это за поля?
2. Какова длина (в байтах) каждого поля заголовка?
3. Длина чего указана в поле «Length»?
4. Какова максимальная длина поля данных UDP?
5. Какой максимально возможный номер порта источника?
6. Укажите номер протокола для UDP (см. соответствующее поле IP-дейтаграммы) в 10-чном и 16-чном виде.

Задание 9. TCP

- 1) запустить *Wireshark*;
- 2) запустить процесс захвата трафика;
- 3) открыть любой сайт; в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;
- 4) остановить захват трафика;
- 5) настроить фильтр просмотра для просмотра сегментов tcp.

Пояснить каким образом происходит начало TCP-сессии (процесс трехэтапного рукопожатия)

Варианты индивидуальных заданий

Варианты индивидуальных заданий

Номер варианта	Номера заданий		
1	1	3	5

2	6	2	7
3	8	4	1
4	9	7	8
5	4	5	6
6	9	6	3
7	8	5	9
8	2	3	4
9	5	7	8
10	3	9	4
11	7	1	3
12	1	8	9
13	2	6	8
14	3	6	2
15	6	4	5

Содержание отчета по индивидуальной практической работе

№ 1

Индивидуальная практическая работа № 1 должна содержать:

- титульный лист;
- Номер варианта условие индивидуального задания;
- Ответы на вопросы заданий с поясняющими скриншотами программы

Wireshark.