Золотых М.О., версия 2023.04

ЗНАКОМСТВО С СЕТЕВЫМ АНАЛИЗАТОРОМ WIRESHARK

Описание работы

При диагностике сетевых проблем и изучении поведения сетевых приложений бесценным источником информации являются данные анализа трафика. Приложения, которые позволяют работать с трафиком называют **сетевыми анализаторами** или **сниферами** (от анг. *to sniff* – вынюхивать). Анализаторы трафика позволяют перехватывать все пакеты, проходящие через сетевую карту, и представлять их в «сыром» (как последовательность битов и байтов) или уже декодированном на параметры и флаги виде. Одним из самых распространённых сетевых анализаторов является **Wireshark** (очень старое название Ethereal). Данная работа посвящена знакомству с этим приложением и изучению приёмов анализа трафика.

Wireshark распространяется под свободной лицензией GNU GPL 2 [1]. Официальный сайт проекта **https://wireshark.org/**. Одной из особенностей приложения является постоянно пополняющаяся база декодера пакетов. Приложение автоматически распознаёт формат пакетов для сотен протоколов от канального до прикладного уровня. При наличии соответствующего аппаратного обеспечения Wireshark может быть использован для анализа трафика Wi-Fi, Bluetooth, GSM, LTE, USB и т.д.

Ещё одним достоинством Wireshark является наличие открытого исходного кода программы и наличие реализаций для Windows, MacOS, Ubuntu и множества Unix-подобных систем.

Программа очень часто обновляется. И с выходом новых версий некоторые элементы интерфейса могут меняться, но, как правило, принципиальных изменений не происходит. Установить Wireshark можно скачав дистрибутив с официального сайта. Во время установки необходимо разрешить инсталляцию библиотеки WinPCap (libpcap в Linix), которая реализует непосредственно захват пакетов.

Во время выполнения работы закройте все лишние вкладки в используемых браузерах и старайтесь не запускать лишних приложений, генерирующих сетевой трафик. Это необходимо для того, чтобы упростить поиск необходимых пакетов. В процессе работы не забывайте сохранять захваченный трафик: он может повторно понадобиться при оформлении отчёта.

Данная работа, с небольшими изменениями, может быть выполнена в ОС семейства Linux.

Системные требования:

- компьютер с ОС Windows 7 или выше;
- подключение к Интернет (Ethernet или Wi-Fi с присвоением адреса по DHCP);
- программа Wireshark версии не ниже 2.6.0 (с установленным пакетом WinPCap), рекомендуется использование англоязычного интерфейса (т.к. он меньше подвержен изменениям).

1. Знакомство с интерфейсом приложения

В начале работы с программой необходимо выбрать сетевой интерфейс, на котором будет производиться захват пакетов. Сделать это можно либо на стартовом экране приложения, либо в окне выбора сетевых интерфейсов. Для этого в меню выберите «*Capture* → *Options*», затем в открывшемся окне отметьте необходимые сетевые адаптеры (при помощи клавиши «*Ctrl*» можно выбрать несколько интерфейсов одновременно) и нажмите кнопку «*Start*». Если необходимые сетевые интерфейсы в списке отсутствуют, закройте программу и запустите ее снова с правами администратора.

Основное рабочее окно Wireshark разделяется на 3 области, как показано на рисунке 0.1. Вы попадёте в это окно сразу, как только запустите захват трафика. В области обозначенной номером 1 отображаются захваченные сетевые пакеты. В области под номером 2 расположен декодер пакетов. В нём содержимое выбранного в первой области пакета разбито по категориям в зависимости от уровня модели OSI. В нижней области, отмеченной номером 3, отображается побайтовое представление этого же пакета. В области номер 4 находятся кнопки для управления захватом пакетов.

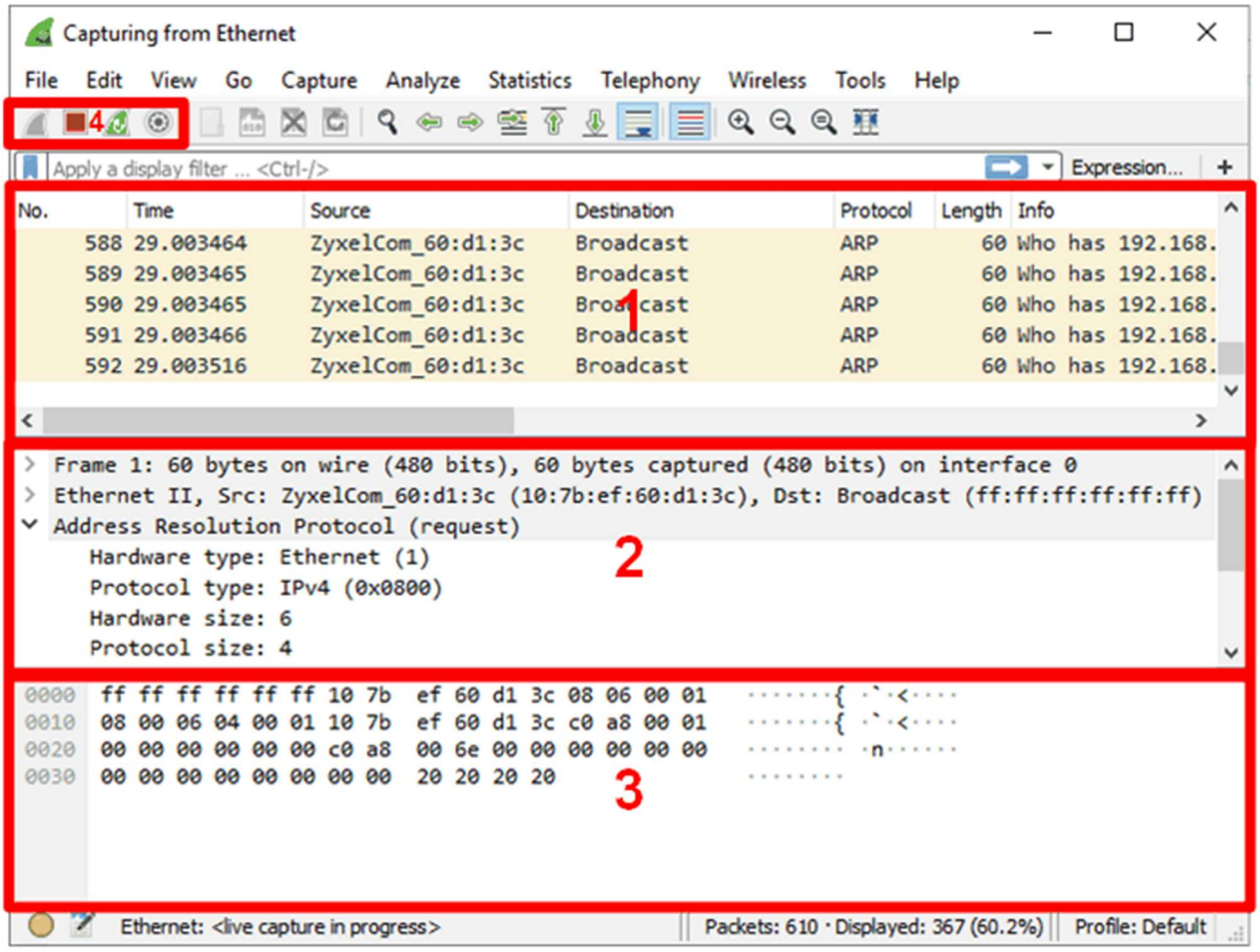

Рисунок 0.1. рабочее окно Wireshark

При запущенном захвате трафика область номер 1 автоматически прокручивается вниз (туда, где появляются новые пакеты). Автопрокрутка включается/выключается при помощи кнопки, отмеченной на рисунке 0.2

Рисунок 0.2. Кнопка включения/отключения автопрокрутки области захвата пакетов

Обратите внимание, что захват пакетов нельзя поставить на паузу. Его можно только остановить. Но при следующем запуске захвата пакетов все предыдущие данные будут очищены. Чтобы избежать потери данных, захваченные пакеты могут быть сохранены в файле, а затем загружены оттуда (эти операции доступны из меню «*File*»). Файл, в который записаны захваченные пакеты называют **дампом трафика**.

Стандартный набор столбцов в списке захваченных пакетов может оказаться неудобным для некоторых задач анализа. Его можно изменить. Для этого необходимо вызвать окно настроек при помощи пункта меню «*Edit → Preferences*». В открывшемся окне выбрать раздел «*Appearance → Columns*», в котором можно удалить ненужные столбы (кнопка «-») или добавить новые (кнопка «+»).

При добавлении столбца необходимо указать его имя и выбрать отображаемую в нём информацию. На рисунке 0.3 приведён пример создания столбца для отображения порта-источника (для транспортного протокола).

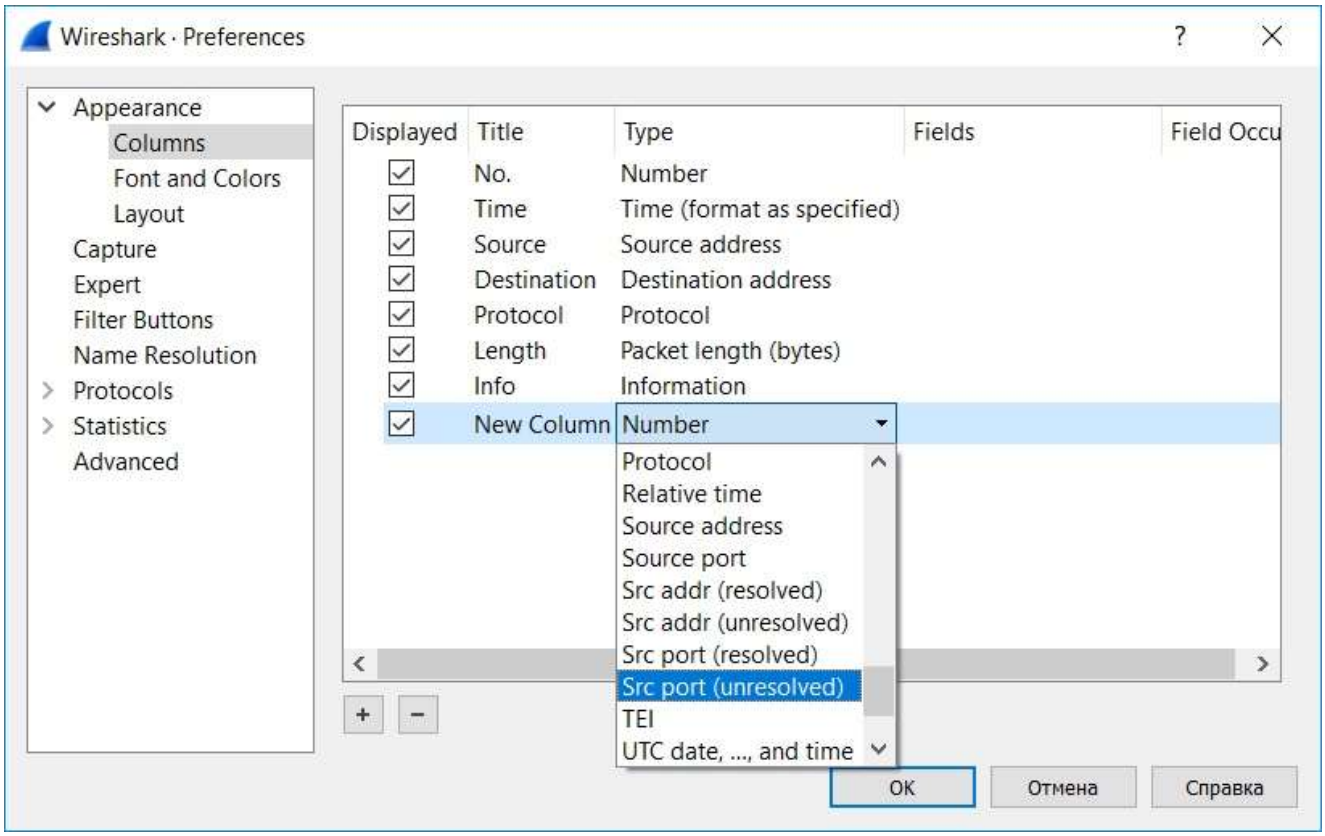

Рисунок 0.3. Пример добавления столбца в список захваченных пакетов

Для того, чтобы отрегулировать ширину столбца можно воспользоваться его контекстным меню в основном окне приложения и выбрать там пункт «*Resize To Contents*».

1.1. Запустите Wireshark. Перечислите доступные для захвата трафика интерфейсы. С использованием утилит **ipconfig** и **route**, определите, какой из интерфейсов используется для отправки пакетов шлюзу по умолчанию. Начните захват трафика для этого интерфейса (в дальнейшей работе также используйте этот интерфейс).

1.2. Продолжайте захват трафика в течении минуты, затем остановите его. Просмотрите захваченные пакеты. Перечислите, к каким прикладным протоколам они относятся. Для протоколов, которые не рассматривались в теоретической части курса, самостоятельно найдите информацию об их назначении.

1.3. Выберете один любой из захваченных пакетов. В декодере пакетов последовательно раскройте информацию обо всех вложенных друг в друга протоколах. Зафиксируйте результат на снимке экрана. Как вы считаете, какое приложение и для чего сформировало данный пакет?

1.4. В списке захваченных пакетов добавьте столбцы, содержащие порт источника и порт назначения. Измените формат отображения времени в столбце

«*Time*» с секунд, прошедших с начала захвата, на абсолютное время с точностью до миллисекунд. Для этого выберите пункты меню «*View → Time Display Format → Date and Time of Day*» и «*View → Time Display Format → Milliseconds*». Отрегулируйте ширину всех столбцов так, чтобы не возникало больших пустых мест между ними и в то же время содержимое не обрезалось (кроме столбца «*Info*»). Зафиксируйте результат на снимке экрана.

1.5. Сохраните полученный трафик при помощи меню «*File*». Какое расширение и размер имеет полученный файл?

2. Фильтрация пакетов

Очень редко для анализа нужны все пакеты. Чаще всего интерес представляет только часть из них. Для того, чтобы оставить только существенную информацию предназначен механизм фильтрации. Фильтры в Wireshark бывают двух типов:

- **фильтры захвата** (capture filters);
- **фильтры отображения** (display filters).

Как видно из названия, в первом случае фильтр ограничивает пакеты, которые будут помещены в память приложения, т.е. часть трафика теряется. Во втором случае ограничения действуют только на список отображаемых пакетов в главном окне, при этом все захваченные пакеты остаются в памяти. Таким образом, фильтр отображения можно многократно изменять без потери данных. Например, на одном и том же дампе можно сначала отобразить все ARP-пакеты, а затем только ICMP. Напротив, если фильтр захвата был настроен только на ARP-пакеты, то все остальные пакеты не были зафиксированы и восстановить их невозможно. Фильтр отображения удобнее использовать, но хранение всех пакетов требует большого объёма оперативной памяти. Например, если понадобится анализировать поток пакетов на гигабитных скоростях, то при записи всего трафика оперативная память обычного ПК может быть израсходована всего за несколько секунд или минут. В этом случае необходимо использовать фильтр захвата, ограничившись только какими-то определёнными пакетами, вместо всех подряд.

Фильтры представляют собой логическое выражение, состоящее из параметров пакета, их значений и логических операторов «И», «ИЛИ» и «НЕ». К сожалению, синтаксис фильтров захвата и отображения существенно отличается. Это обусловлено тем, что в первом случае фильтр отправляется в WinPcap, а во втором случае во встроенный декодер пакетов Wireshark.

Фильтр захвата более примитивен (меньше набор параметров), чем фильтр отображения, но это объясняется требованиями, предъявляемыми к его быстродействию. В качестве логических операторов в нём используются «*and*», «*or*» и «*not*» (как в языке Python). Для группировки логических операторов используются скобки. Значения параметров записываются сразу после параметра через

пробел без знака сравнения. Подпараметры также записываются после параметра через пробел. Например, «*ip proto 0x11 and (port 53 or port 67)*» соответствует правилу «*захватывать только UDP трафик, использующий порты 53 и 67*» или, иными словами, «*захватывать только DNS и DHCP пакеты*». Более подробно с синтаксисом фильтра можно ознакомиться на сайте проекта WinPacp [2].

Чтобы создать новый фильтр, в меню выберете пункт «*Capture → Capture Filters…*», далее в открывшемся окне нажмите на кнопку «+», а затем в появившейся строке впишите имя фильтра (произвольное) и критерий фильтрации. Пример создания фильтра захвата приведён на рисунке 0.4.

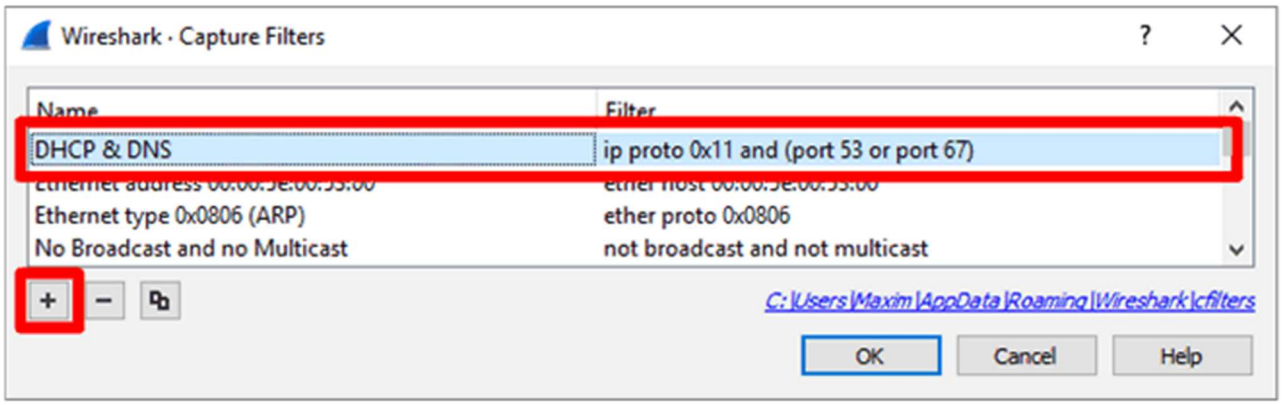

Рисунок 0.4. Создание фильтра захвата

Применить созданный фильтр можно перед началом захвата трафика. Выпадающий список для выбора фильтра доступен как на стартовом экране (выше списка интерфейсов), так и на экране в окне выбора сетевых интерфейсов, который вызывается пунктом меню «*Capture → Options*» (здесь выбор фильтра находится ниже списка интерфейсов, рис. 0.5).

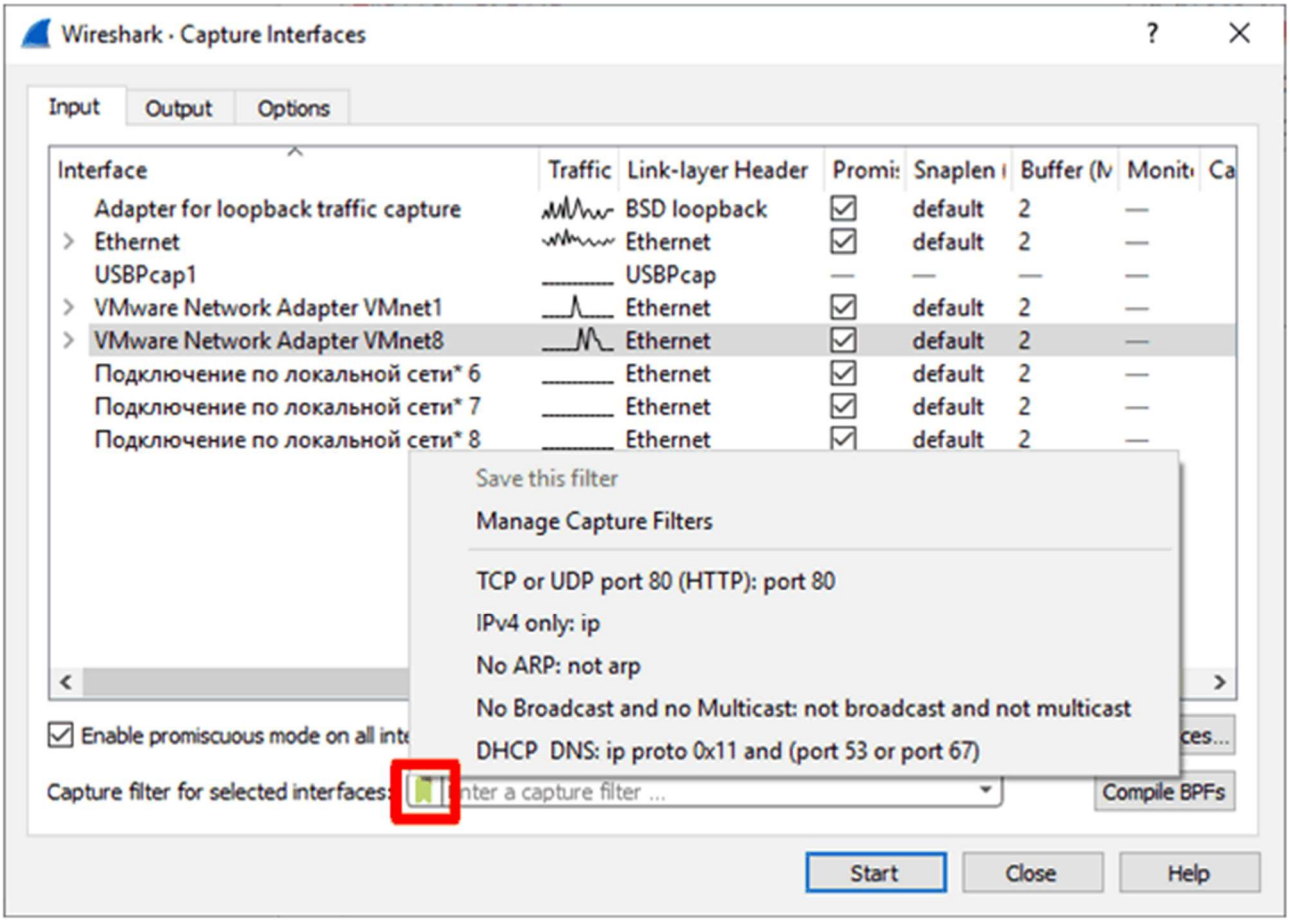

Рисунок 0.5. Выбор фильтра захвата

Фильтр отображения обладает синтаксисом, схожим с языком программирования C: в качестве логических операторов используются «&&», «||» и «*!*», операторы сравнения используются «==», «*!=*», «<», «>» и др., подпараметры записываются через «.» после параметра. Предыдущему критерию выборки DNS и DHCP пакетов для фильтра отображения соответствует: «*udp.port==53 || udp.port==67*». А благодаря богатому словарю декодера пакетов можно сразу записывать фильтр в виде «*dns || dhcp*».

Фильтр отображения вводится в специальную строку, находящуюся сразу над списком захваченных пакетов. Он может быть использован как в процессе захвата пакетов, так и после.

Для упрощения процедуры написания фильтров отображения можно автоматически конструировать фильтр по любому полю в декодере пакетов. Для этого необходимо найти интересующий пакет в списке и выделить его. Затем, когда его содержимое отобразится в декодере пакетов, вызвать контекстное меню для интересующего поля и выбрать там пункт «*Apply as Filter → Selected*» или «*Apply as Filter → ... and Selected*» (если необходимо уточнить уже заданную выборку). На рисунке 0.6 приведён пример вы создания фильтра для ICMP эхо-ответа. В результате в поле фильтра появится выражение «*icmp.type == 0*».

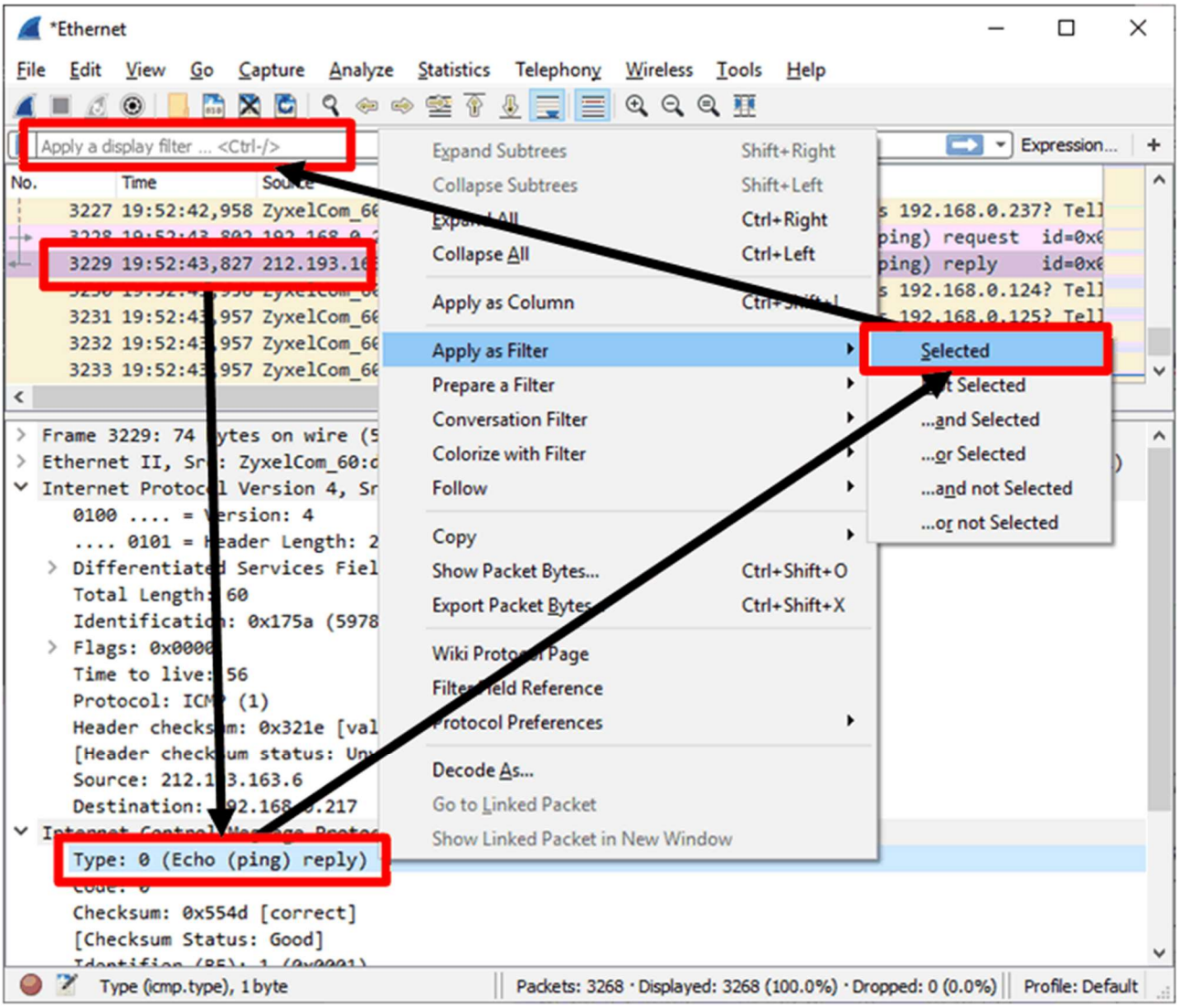

Рисунок 0.6. Создание фильтра отображения для ICMP эхо-ответов

В системе встроено множество фильтров отображения, но, если вам их не хватает (например, при работаете с собственным или очень редкими протоколами), то в условиях фильтра можно обращаться напрямую к значениям байт по их номеру. Для этого необходимо использовать запись вида: «*frame[X:Y]==Z*», где *X* – смещение относительно начала данных (считая с 0), *Y* – количество от заданного смещения и *Z* – искомое значение в шестнадцатеричном виде. Вместо «*frame*», который соответствует отсчёту от канальной части Ethernet-кадра, можно использовать названия вышестоящих протоколов. В этом случае смещение будет отсчитываться относительно начала заголовка данного протокола в пакете.

Например, запись «*frame[23:1]==01*» будет соответствовать протоколу ICMP, так как 23 байт от начла Ethernet-кадра может содержать поле «*Protocol*», но только в том случае, если в него вложен IPv4 протокол. Если внутри кадра окажется IPv6 протокол, то в этом байте будет находиться фрагмент IP-адреса источника. А если протокол CDP, то результат будет ещё более непредсказуем. Таким образом, фильтр «*frame[23:1]==01*» действительно позволит получить

все ICMP сообщения, но кроме них в выборку попадёт и псевдослучайная информация. Поэтому логическое выражение желательно уточнить, дополнив выборкой только IPv4: «*frame[23:1]==01 && frame[12:2]==0800*» или используя смещения от IPv4 заголовка: «*ip[9:1]==01*». Здесь смещение 9 от начала IPv4 заголовка соответствует смещению 23 от Ethernet заголовка так как заголовок Ethernet DIX имеет длину 14 байт.

2.1. Запустите захват трафика в Wireshark и перейдите в веб-браузере на любой сайт по вашему выбору. Остановите захват пакетов. При помощи декодера пакетов создайте фильтр для отображения только DNS пакетов. Зафиксируйте результат на снимке экрана.

2.2. Очистите фильтр и снова выберите любой DNS-пакет из списка захваченных. При помощи декодера пакетов в заголовке UDP найдите поле, содержащее номер порта «*53*» (соответствует протоколу DNS), и используйте значение этого поля в качестве нового фильтра. Зафиксируйте результат на снимке экрана.

2.3. Чем отличается результаты фильтрации в пунктах 2.1 и 2.2? Что необходимо исправить в фильтре из пункта 2.2, чтобы, не используя ключевого слова «*dns*», всё-таки выбрать все DNS-пакеты?

2.4. Какими узлам адресованы DNS пакеты? Найдите эти IP-адреса в выводе утилиты «**ipconfig /all**» и отметьте их на снимке экрана.

2.5. Создайте фильтр захвата, который будет пропускать только ARP и DNS пакеты (не забудьте, что клиентский DNS определяется не только 53 портом, но и транспортом UDP). Начните захват трафика и продолжайте его до тех пор, пока не будут захвачены пакеты двух искомых типов (это не должно занять больше нескольких минут). Пользуясь приёмами из работы **№Ошибка! Источник ссылки не найден.** вы можете спровоцировать появление данных пакетов. Зафиксируйте результат на снимке экрана и запишите фильтр, который вы использовали.

2.6. Составьте фильтр отображения для выборки ARP-запросов, используя обращение к байтам по их смещению относительно начала Ethernet кадра. Узнать необходимое смещение (тип ARP пакета) можно из документов RFC, либо исследовав образец пакета (когда вы выбираете интересующий параметр в декодере пакета, соответствующие байты в «сыром» представлении выделяются цветом). Запишите полученный фильтр.

2.7. Запустите захват трафика, используя фильтр захвата, пропускающий только DHCP пакеты (обратитесь к теоретическому материалу за информацией об используемых портах и транспортном протоколе). Запишите полученный фильтр

2.8. В командной строке, с правами администратора, выполните команды «**ipconfig /release**» и «**ipconfig /renew**», которые заставляют ОС сначала отказаться от динамического адреса, а затем запросить его вновь. Остановите захват трафика. Зафиксируйте результат на снимке экрана. Какие типы DHCP пакетов вы захватили? Какова роль каждого из этих типов?

2.9. В декодере пакетов изучите пакет DHCPOFFER. Какие сетевые настройки в нём предложены клиенту?

3. Анализ служебных протоколов

3.1. Запустите Wireshark и начните захват трафика. Запустите командную строку от имени администратора и очистите ARP-таблицу рабочего места командой «**arp -d ***». Выполните эхо-тестирование сначала шлюза по умолчанию (можно определить при помощи утилиты **ipconfig**), а затем узла **yandex.ru**. Завершите захват трафика. Чтобы скрыть ненужный трафик, установите фильтр для отображения только ARP, DNS и ICMP-пакетов. Зафиксируйте результат на снимке экрана.

3.2. Сколько ARP-пакетов и какого типа было захвачено? Какие узлы были отправителями и получателями этих пакетов (определите по MAC-адресам, что это за узлы)? Какое отношение эти пакеты имели к эхо-тестированию? Какое значение в поле *EtherType* заголовка Ethernet указывает на наличие протокола ARP во вложении?

3.3. Сколько DNS-пакетов и какого типа было захвачено? Какие узлы были отправителями и получателями этих пакетов на канальном и сетевом уровнях? Какое отношение эти пакеты имели к эхо-тестированию?

3.4. Среди захваченных ICMP пакетов несколько запросов и несколько ответов. Выберите любой эхо-запрос в списке захваченных пакетов. Зафиксируйте на снимке экрана содержимое декодера для заголовков IP и ICMP выбранного пакета. При выборе пакета, слева от его номера в общем списке появляется символ «→». Найдите пакет, отмеченный парной стрелкой «←». Это ответный пакет. Зафиксируйте на снимке экрана содержимое декодера пакетов для его заголовков IP и ICMP. На двух полученных снимках экранов отметьте поля, которые указывают на то, что эти пакеты взаимосвязаны.

3.5. Создайте фильтр, который позволит отобразить только ICMP пакеты, которые использовались для эхо-тестирования узла **yandex.ru**. Запишите этот фильтр. Выпишите IP и MAC-адреса получателей и отправителей пакетов для эхо-запроса. Каким узлам принадлежат эти адреса? Почему один и тот же пакет на сетевом и канальном уровнях адресован различным узлам?

3.6. Выполните эхо-тестирование узла **yandex.ru** пакетом длиной 1400 байт с захватом трафика. Чем заполнено поле данных и какую длину оно имеет в запросах и ответах? Выполните эхо-тестирование узла **google.com** пакетом длиной 1400 байта с захватом пакетов. Чем заполнено поле данных и какую длину оно имеет в запросах и ответах? Обратите внимание, что во втором случае Wireshark напротив эхо-запроса ставит пометку «*no response found!*», хотя ответ существует и распознаётся утилитой **ping**. Это объясняется тем, что эти приложения проверяют разный набор полей при поиске парного пакета.

#Дополнительно

Почти все ОС для своей реализации ping позволяют задать длину пакета. Однако они по-разному интерпретируют этот параметр. Так, например, Windows под длиной пакета понимает длину поля данных ICMP пакета (для MTU 1500 без фрагментирования может варьироваться от 0 до 1472). А Cisco IOS под длиной пакета поднимает полную длину IP-пакета (для MTU 1500 без фрагментирования может варьироваться от 28 до 1500).

3.7. Выполните эхо-тестирование узла **yandex.ru** пакетом длиной 3000 байт с захватом трафика. Если вы используете фильтр отображения «*icmp*», то из отображения пропадёт часть IP-пакетов, содержащих фрагменты эхо-запроса. Почему так происходит? Используйте фильтр с использованием условия «*ip.addr==...*», чтобы увидеть все фрагменты интересующего нас эхо-тестирования. Для любого из эхо-запросов найдите все три его фрагмента. Выпишите значения полей «*Identification*», «*MF*», «*Fragment offset*» IP-заголовка для этих пакетов. Поясните, каким образом значения этих полей позволяют восстановить порядок следования фрагментов и определить, их число.

3.8. Выполните эхо-тестирование узла **yandex.ru** с ключом «**-r 9**» с захватом трафика. Найдите в какой части сетевого пакета содержится полученная вами информация о маршруте? Объясните, почему 9 – максимально возможное значение опции «**-r**».

3.9. Создайте фильтр захвата трафика, который пропускает только ICMP и DNS-пакеты (запишите его). Начните захват трафика с созданным фильтром, а затем выполните трассировку узла **yandex.ru**. По завершении трассировки остановите захват пакетов. Зафиксируете результат захвата на снимке экрана.

3.10. Какие типы ICMP сообщений были захвачены? Какие узлы сообщения каких типов отправляют? Почему различные узлы присылают сообщение с типом 11 в ответ запрос к одному и тому же узлу?

3.11. Отфильтруйте пакеты, оставив только эхо-запросы (выберите в декодере поле, определяющее типа пакета, и используйте его как фильтр). Почему, несмотря на фильтр по типу 8, в списке по-прежнему отображаются ICMP-сообщения типа 11? Каким образом можно усовершенствовать фильтр, чтобы оставить только исходные запросы?

3.12. В списке захваченных пакетов есть DNS-пакеты. Какова их роль в процессе трассировки? Какой тип DNS-записи в них используется?

4. Обращение к веб-сайту

4.1. Запустите захват трафика в Wireshark и откройте сайт **http://linuxatemyram.ru** (ни в коем случае не используйте схему **https**). Когда сайт полностью откроется – остановите захват трафика. При помощи команды «**ipconfig /displaydns**» определите, какой IP-адрес был ис-

пользован для запроса содержимого этого сайта и используйте его в качестве значения параметра «**ip.addr**» фильтра отображения. Зафиксируйте результат на снимке экрана.

4.2. В списке захваченных пакетов найдите тот, у которого текст в столбце «*Info*» начинается с «*GET / HTTP …*». Вызовите контекстное меню для этого пакета и выберите пункт «*Follow → TCP Stream*». Это позволит отобразить данные, переданные во всех пакетах, относящихся к одному потоку. Красным цветом выделены данные, переданные клиентом, а синим – сервером. Если ответ сервера начинается с «*HTTP/1.1 304 …*», то это указывает, что страница была загружена не с сервера, а из локального кэша (возможно вы не с первого раза смогли записать трафик или ранее посещали данный сайт). В этом случае вернитесь назад и повторите пункт 4.1, обновив страницу в браузере при помощи сочетания клавиш *Ctrl+F5* (принудительное обновление кэша). Зафиксируйте содержимое TCP-потока на снимке экрана.

4.3. Какое значение принял фильтр отображения, когда вы выбрали все сообщения, относящиеся к одному потоку? Изучите первые три пакета в потоке: какие флаги у них установлены? Определите, в каких пакетах (теперь для всего потока) изменяются опции «*MSS*» и «*Windows Scale*» и какое значение они принимают. С учётом информации о множителе определите размер окна и изменяется ли он в течении потока. Обратите внимание, что размер окна, его множитель и «*MSS*» изменяются независимо для серверной и для клиентской стороны.

4.4. Для всех остальных пакетов определите: размеры окон: принимаемого сервером и клиентом.

4.5. Выпишите все заголовки, использованные в HTTP запросе и ответе. Самостоятельно найдите и внесите в отчёт информацию о том, какой из этих заголовков для чего предназначен.

4.6. Очистите фильтры отображения и убедитесь, что захват пакетов остановлен. Воспользуйтесь пунктом меню «*File → Export Objects → HTTP*», вызывающим встроенный декодер для прикладного протокола. Какие файлы присутствуют в списке? Какова их роль в формировании исследуемой страницы? Найдите среди файлов изображение в формате **png** и сохраните его на компьютере (выберите файл и воспользуйтесь кнопкой «*Save As…*»). Откройте сохранённое изображение. Зафиксируйте результат на снимке экрана.