АНАЛИЗАТОРЫ СЕТЕВЫХ ПРОТОКОЛОВ

Данная тема посвящена знакомству с программным обеспечением, предназначенным для анализа сетевого трафика на примере бесплатного и одного из самых мощных и удобных анализаторов трафика - WireShark.

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

Во время работы сниффера сетевой интерфейс переключается в т. н. «режим прослушивания» (promiscuous mode), что и позволяет ему получать пакеты, адресованные другим интерфейсам в сети.

Методы перехвата трафика:

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

В начале 1990-х сниффинг широко применялся хакерами для захвата пользовательских логинов и паролей, которые в ряде сетевых протоколов передаются в незашифрованном или слабо-зашифрованном виде. Широкое распространение концентраторов позволяло захватывать трафик без больших усилий в больших сегментах сети практически без риска быть обнаруженным.

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

1. Обнаружить паразитный, вирусный и закольцованный трафик, наличие которого увеличивает загрузку сетевого оборудования и каналов связи (снифферы здесь малоэффективны; как правило, для этих целей используют сбор разнообразной статистики серверами и активным сетевым оборудованием и её последующий анализ).

2. Выявить в сети вредоносное и несанкционированное ПО, например, сетевые сканеры, флудеры, троянские программы, клиенты пиринговых сетей и другие (это обычно делают при помощи специализированных снифферов — мониторов сетевой активности).

3. Перехватить любой незашифрованный (иногда «полезен» и зашифрованный) пользовательский трафик с целью получения паролей и другой информации.

4. Локализовать неисправность сети или ошибку конфигурации сетевых агентов (для этой цели снифферы часто применяются системными администраторами)

Поскольку в «классическом» снифферe анализ трафика происходит вручную, с применением лишь простейших средств автоматизации (анализ протоколов, восстановление TCP-потока), то он подходит для анализа лишь небольших его объёмов.

Снизить угрозу сниффинга пакетов можно с помощью таких средств как аутентификация, криптография, антиснифферы, коммутируемая инфраструктура.

Снифферы можно разделить на категории:

- анализаторы протоколов (Wireshark, Network Miner, TracePlus32 Web Detective, CommView);
- пакетные снифферы (RawCap, tcpdump, Network Probe, Etherscan Analyzer).
- снифферы беспроводных сетей (Kismet, airodump-ng, CommView for WiFi), перехватывают трафик беспроводных сетей даже без подключения к этим сетям;
- парольные снифферы (Cain & Abel, Ace Password Sniffer), перехватывают и контролируют разнообразные пароли;
- HTTP снифферы (HTTP Analyzer, IEWatch Professional, EffeTech HTTP Sniffer), перехватывают HTTP заголовки;
- принт снифферы (O&K Print Watch, PrintMonitor, Print Inspector), позволяют контролировать и управлять процессом печати в сети;
- снифферы IM систем (MSN Shiffer, ICQ Sniffer, AIM Sniff, IM-Sniffer), предоставляют перехваченную переписку в удобно читаемом виде;

1. Принципы работы снифферов.

Сниффер — это программа, которая работает на уровне сетевого адаптера NIC (Network Interface Card) (канальный уровень) и скрытым образом перехватывает весь трафик. Поскольку снифферы работают на канальном уровне модели OSI, они не должны играть по правилам протоколов более высокого уровня. Снифферы обходят механизмы фильтрации (адреса, порты и т.д.), которые драйверы Ethernet и стек TCP/IP используют для интерпретации данных. Пакетные снифферы захватывают из провода (эфира) все, что по нему приходит. Снифферы могут сохранять кадры в двоичном формате и позже расшифровывать их, чтобы раскрыть информацию более высокого уровня, спрятанную внутри (рис. 1).

Promiscuous mode.

Для того чтобы сниффер мог перехватывать все пакеты, проходящие через сетевой адаптер, драйвер сетевого адаптера должен поддерживать режим функционирования promiscuous mode (режим прослушивания). Данный режим работы сетевого адаптера автоматически активизируется при запуске сниффера или устанавливается вручную соответствующими настройками сниффера (требуются права суперпользователя).

Весь перехваченный трафик передается декодеру пакетов, который идентифицирует и расщепляет пакеты по соответствующим уровням иерархии. В зависимости от возможностей конкретного сниффера представленная информация о пакетах может впоследствии дополнительно анализироваться и отфильтровываться.

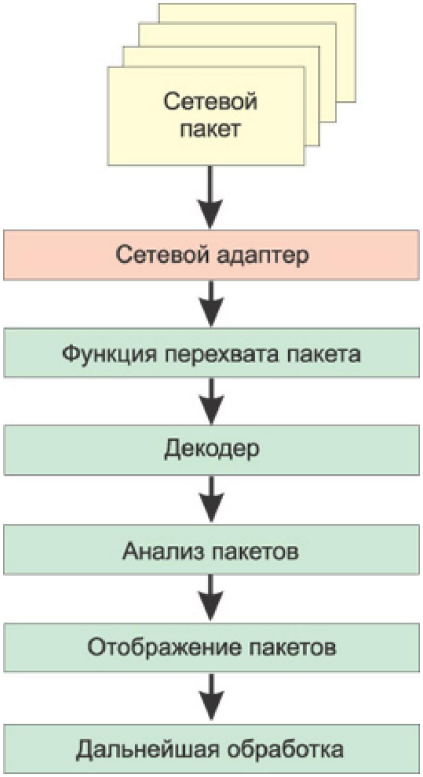


Рис. 1. Схема работы сниффера.

2. Ограничения использования снифферов.

Значительную опасность снифферы представляли в те времена, когда информация передавалась по сети в открытом виде (без шифрования), а локальные сети строились на основе концентраторов (хабов). Однако эти времена безвозвратно ушли, и в настоящее время использование снифферов для получения доступа к конфиденциальной информации — задача отнюдь не из простых.

Дело в том, что при построении локальных сетей на основе концентраторов (хабов) существует некая общая среда передачи данных (сетевой кабель) и все узлы сети обмениваются пакетами, конкурируя за доступ к этой среде (рис. 2), причем пакет, посылаемый одним узлом сети, передается на все порты концентратора и этот пакет прослушивают все остальные узлы сети, но принимает его только тот узел, которому он адресован. Если на одном из узлов сети установлен пакетный сниффер, то он может перехватывать все сетевые пакеты, относящиеся к данному сегменту сети (сети, образованной концентратором).

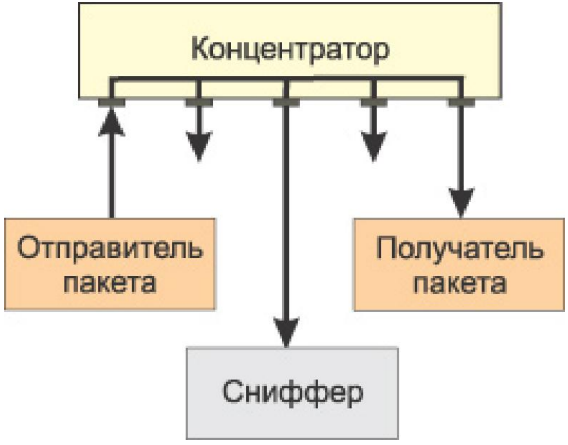


Рис. 2. При использовании концентраторов сниффер способен перехватывать все пакеты сетевого сегмента.

Коммутаторы являются более интеллектуальными устройствами, чем широковещательные концентраторы, и **изолируют сетевой трафик**. Коммутатор знает адреса устройств, подключенных к каждому порту, и передает пакеты только между нужными портами. Это позволяет разгрузить другие пор- ты, не передавая на них каждый пакет, как это делает концентратор. Таким образом, посланный неким узлом сети пакет передается только на тот порт коммутатора, к которому подключен получатель пакета, а все остальные узлы сети не имеют возможности обнаружить данный пакет (рис. 3).

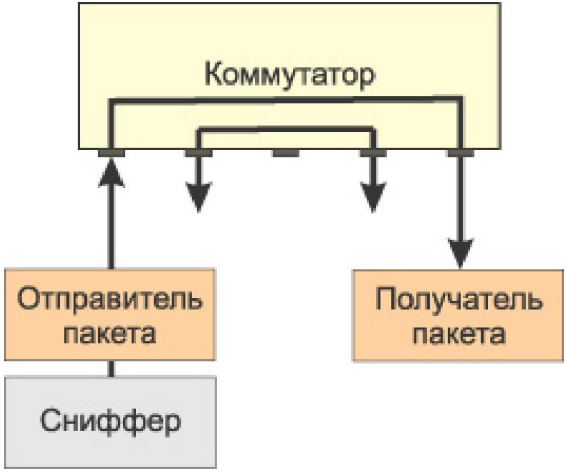


Рис. 3. При использовании коммутаторов сниффер способен перехватывать только входящие и исходящие пакеты одного узла сети.

Поэтому если сеть построена на основе коммутатора, то сниффер, установленный на одном из компьютеров сети, способен перехватывать только те пакеты, которыми обменивается данный компьютер с другими узлами сети. В результате, чтобы иметь возможность перехватывать пакеты, которыми интересующий злоумышленника компьютер или сервер обменивается с остальными узлами сети, необходимо установить сниффер именно на этом компьютере (сервере), что на самом деле не так-то просто.

Другая причина, по которой снифферы перестали быть настолько опасными, как раньше, заключается в том, что в настоящее время наиболее важные данные передаются **в зашифрованном виде.** Открытые, незашифрованные службы быстро исчезают из Интернета. К примеру, при посещении web-сайтов все чаще используется протокол SSL (Secure Sockets Layer); вместо открытого FTP используется SFTP (Secure FTP), а для других служб, которые не применяют шифрование по умолчанию, все чаще используются виртуальные частные сети (VPN).

Но, есть пакетные снифферы запускающиеся из командной строки и **не имеющие графического интерфейса (tcpdump).** Такие снифферы, в принципе, можно устанавливать и запускать удаленно и скрытно для пользователя.

Но, большинство управляемых коммутаторов имеют **функцию зеркалирования портов**. То есть порт коммутатора можно настроить таким образом, чтобы на него дублировались все пакеты, приходящие на другой или несколько других (все) порты коммутатора. Если в этом случае к такому порту подключен компьютер с пакетным сниффером, то он может **перехватывать все пакеты**, которыми обмениваются компьютеры в данном сетевом сегменте. Однако, как правило, возможность конфигурирования коммутатора доступна только

сетевому администратору, что не означает, что он не может быть злоумышленником.

Но, применение криптографии базируется на сертификатах (ключах) (Thawte, VeriSign) и существуют техники атак (MITM) **для подмены ключей шифрования** и вскрытия трафика.

Но, снифферы **остаются действенным и мощным средством для диагностирования сетей** и локализации сетевых проблем**.**

Но, снифферы могут с успехом использоваться для аудита сетевой безопасности для:

- обнаружения несанкционированного трафика,

- идентификации несанкционированного программного обеспечения в сети,

- идентифицикации неиспользуемых протоколов засоряющих сеть для их удаления из сети,

- генерации и анализа трафика при испытаниях на вторжение (penetration test) с целью проверки систем защиты,

- совместной работы с системами обнаружения вторжений (Network Intrusion Detection System, NIDS).

3. Общий обзор программных пакетных снифферов.

Все программные снифферы можно условно разделить на две категории: снифферы, поддерживающие запуск **из командной строки**, и снифферы, имеющие **графический интерфейс**. При этом необходимо отметить, что существуют снифферы, которые объединяют в себе обе эти возможности. Кроме того, снифферы отличаются друг от друга протоколами, которые они поддерживают, глубиной анализа перехваченных пакетов, возможностями по настройке фильтров, а также возможностью совместимости с другими программами.

Обычно окно любого сниффера с графическим интерфейсом состоит их трех областей.

В первой из них отображаются итоговые данные перехваченных пакетов. Обычно в этой области отображается минимум полей, а именно: время перехвата пакета; IP-адреса отправителя и получателя пакета; MAC-адреса отправителя и получателя пакета, исходные и целевые адреса портов; тип протокола (сетевой, транспортный или прикладного уровня); некоторая суммарная информация о перехваченных данных.

Во второй области выводится статистическая информация об отдельном выбранном пакете.

В третьей области пакет представлен в шестнадцатеричном виде или в символьной форме — ASCII.

Практически все пакетные снифферы позволяют производить **анализ декодированных пакетов** (именно поэтому пакетные снифферы также называют пакетными анализаторами). Сниффер распределяет перехваченные пакеты по уровням и протоколам. Любой сниффер

способен распознавать протокол TCP, а продвинутые снифферы умеют определять, каким приложением порожден данный трафик. Большинство анализаторов протоколов распознают свыше 500 различных протоколов и умеют описывать и декодировать их по именам.

Одна из проблем, с которой могут сталкиваться анализаторы пакетов, — невозможность корректной идентификации протокола, использующего порт, отличный от порта по умолчанию. К примеру, с целью повышения безопасности вместо традиционного порта 80, зарезервированного для web-сервера, сервер можно принудительно перенастроить на порт 8088 или на любой другой. Некоторые анализаторы пакетов в подобной ситуации не способны корректно определить протокол и отображают лишь информацию о протоколе нижнего уровня (TCP или UDP).

Существуют снифферы с **аналитическими модулями**, позволяющими создавать отчеты с полезной аналитической информацией о перехваченном трафике.

Другая характерная черта большинства программных анализаторов пакетов— возможность **настройки фильтров до и после захвата трафика**. Фильтры выделяют из общего трафика определенные пакеты по заданным критериям, что позволяет при анализе трафика избавиться от лишней информации.

4. Анализатор сетевого трафика WireShark.

4.1. Возможности Wireshark.

Работает на большинстве современных ОС (Microsoft Windows, Mac OS X, UNIX, Linux).

Wireshark – продукт с открытым исходным кодом, распространяемый на основании лицензии GPL. Поэтому сетевое сообщество очень активно добавляет в него поддержку новых протоколов в виде плагинов или напрямую встраивает в исходный код.

Wireshark может перехватывать трафик с различных сетевых устройств, включая беспроводные устройства.

Умеет вести перехват, декодирование и показ трафика сетевого интерфейса в режиме реального времени.

Поддерживает множество (~1000) протокольных декодировщиков (TELNET, FTP, POP, RLOGIN, ICQ, SMB, MySQL, HTTP, NNTP, X11, NAPSTER, IRC, RIP, BGP, SOCKS 5, IMAP 4, VNC, LDAP, NFS, SNMP, MSN, YMSG и другие).

Wireshark позволяет анализировать ранее захваченные пакеты, загрузив их из сохраненного файла.

Основной формат файла Wireshark такой же, как у libpcap.

Wireshark поддерживает импорт и экспорт файлов из других пакетных анализаторов:

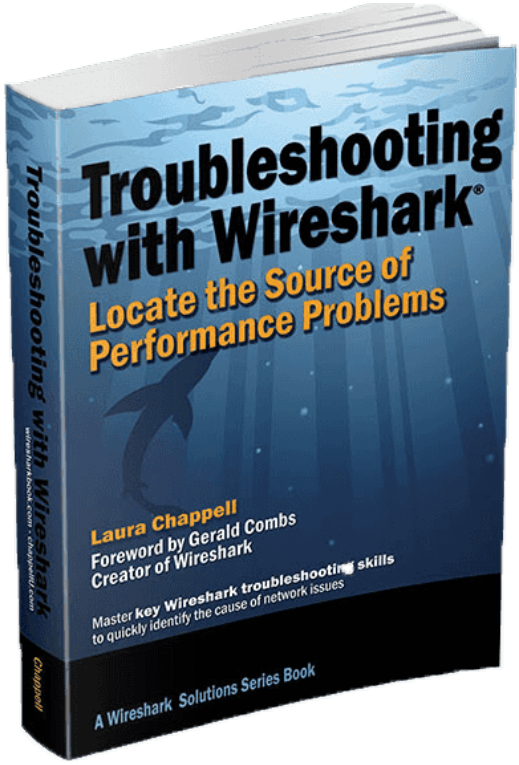
1. libpcap, tcpdump и другие, использующие формат tcpdump
2. SUN snoop и atmsnoop; Shomiti/Finisar Surveyor captures
3. Novell LANalyzer captures
4. Microsoft Network Monitor captures
5. AIX's iptrace captures; Cinco Networks NetXRay captures
6. Network Associates Windows-based Sniffer captures
7. Network General/Network Associates DOS-based Sniffer (compressed/uncompressed) captures
8. AG Group/WildPackets EtherPeek/TokenPeek/AiroPeek/EtherHelp/Packet-Grabber captures
9. RADCOM's WAN/LAN analyzer captures
10. Network Instruments Observer version 9 captures
11. Lucent/Ascend router debug output
12. files from HP-UX's nettl
13. Toshiba's ISDN routers dump output
14. the output from i4btrace from the ISDN4BSD project
15. traces from the EyeSDN USB S0.
16. the output in IPLog format from the Cisco Secure Intrusion Detection System
17. pppd logs (pppdump format)
18. the output from VMS's TCPIPtrace/TCPtrace/UCX$TRACE utilities
19. the text output from the DBS Etherwatch VMS utility
20. Visual Networks' Visual UpTime traffic capture
21. the output from CoSine L2 debug; the output from Accellent's 5Views LAN agents
22. Endace Measurement Systems' ERF format captures
23. Linux Bluez Bluetooth stack hcidump -w traces
24. Catapult DCT2000 .out files

Позволяет фильтровать собираемые пакеты по множеству критерий.

Позволяет искать пакеты по множеству критериев.

Позволяет подсвечивать захваченные пакеты разных протоколов.

Позволяет создавать разнообразную статистику.



Wireshark не умеет:

- Wireshark – это не система обнаружения вторжений. Он не предупредит о том, если кто-то делает странные вещи в сети. Однако если это происходит, Wireshark поможет понять что же на самом деле случилось.

- Wireshark не умеет генерировать сетевой трафик, он может лишь анализировать имеющийся. В целом, Wireshark никак не проявляет себя в сети, кроме как при разрешении доменных имен, но и эту функцию можно отключить.

Для генерации пакетов можно использовать **Packet Excalibur**.

4.2 Установка Wireshark.

Установка сниффера Wireshark под Windows производится мастером установки. Если на компьютере отсутствует библиотека WinPcap, то она будет установлена вместе со сниффером. Для полной установки и использования **promiscuous** моды требуются права суперпользователя. Для анализа готовых pcap-файлов расширенные права не нужны.

Есть Portable Wireshark – для запуска без установки.

На шаге выбора компонентов можно установить некоторые сопутствующие инструменты:

- TShark – консольный анализатор сетевого трафика; - Rawshark – фильтр «сырых» пакетов;

- Editcap – утилита, позволяющая открывать сохраненные пакетные дампы и изменять их;

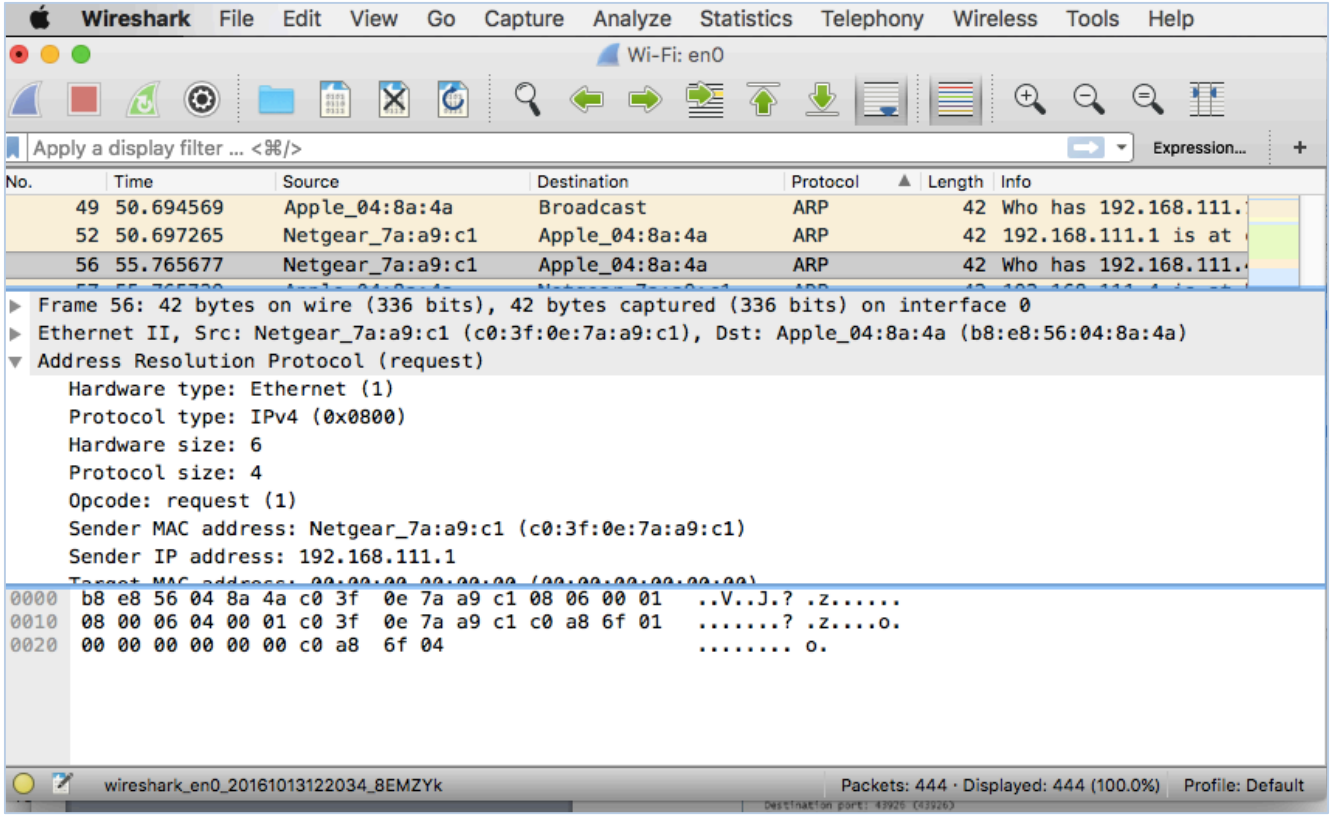
- Text2Pcap – утилита для конвертации HEX-дампов пакетов в формат Pcap;

- Mergecap – утилита для соединения нескольких дампов в один файл;

- Capinfos – утилита для предоставления информации о сохраненных дампах;

- некоторые плагины расширенной статистики.

4.3. Интерфейс Wireshark (рис. 4).



Сверху находятся стандартные строка меню и панель инструментов.

Далее следует окно фильтра, в нем можно задавать критерии фильтрации пакетов или выбирать готовые фильтры, подробнее работу с фильтрами рассмотрим позже.

Следом идет окошко со списком всех перехваченных пакетов. В нем доступна такая информация как: номер пакета, относительное время получения пакета (отсчет производится от первого пакета; параметры отображения времени можно изменить в настройках), IP адрес отправителя, IP адрес получателя, протокол, по которому пересылается пакет, а также дополнительная информация о нем. Разные протоколы подсвечиваются разными цветами, что добавляет наглядности и упрощает анализ.

Далее расположено окно, в котором представлена детальная информация о пакете согласно сетевой модели OSI.

Самое нижнее окно показывает пакет в «сыром» HEX виде, то есть побайтово.

Конфигурация интерфейса может быть легко изменена в меню View. Например, можно убрать окно побайтового представления пакета (оно же Packet Bytes в меню View), так как в большинстве случаев (кроме анализа данных в пакете) оно не нужно и только дублирует информацию из окна детального описания.

4.4. Перехват трафика

Перехват трафика является одной из ключевых возможностей Wireshark.

Движок Wireshark по перехвату предоставляет следующие возможности:

- перехват трафика различных видов сетевого оборудования (Ethernet, Token Ring, ATM и другие);

- прекращение перехвата на основе разных событий: размера перехваченных данных, продолжительность перехвата по времени, количество перехваченных пакетов;

- показ декодированных пакетов во время перехвата;

- фильтрация пакетов с целью уменьшить размер перехваченной информации;

- запись дампов в несколько файлов, если перехват продолжается долго.

Движок не может выполнять следующие функции:

- перехват трафика с нескольких сетевых интерфейсов одновременно (однако, существует возможность запустить несколько копий Wireshark – каждая для своего интерфейса);

- прекращение перехвата в зависимости от перехваченной информации.

4.4.1. Настройка перехвата.

Чтобы начать перехват трафика нужно иметь права Администратора на данной системе и выбрать правильный сетевой интерфейс.

Для выбора сетевого адаптера, с которого будет выполняться перехват нужно воспользоваться меню Capture → Options... Появится окно со списком сетевых интерфейсов, доступных в системе (рис. 5).

В этом окне можно увидеть такую информацию как название интерфейса, IP адрес интерфейса, сетевая активность интерфейса (представлена в виде общего количества пакетов с момента появления окна и количества пакетов в секунду).

Также из этого окна можно посмотреть настройки перехвата (рис. 6).

В настройках перехвата можно изменять такие параметры как фильтрация пакетов, запись дампа в несколько файлов, прекращение перехвата по разным критериям (количество пакетов, количество мегабайт, количество минут), опции показа пакетов, разрешение имен. В большинстве случаев эти параметры можно оставить по умолчанию.

Возможна фильтрация пакетов с использованием окна Filter (рис. 7).

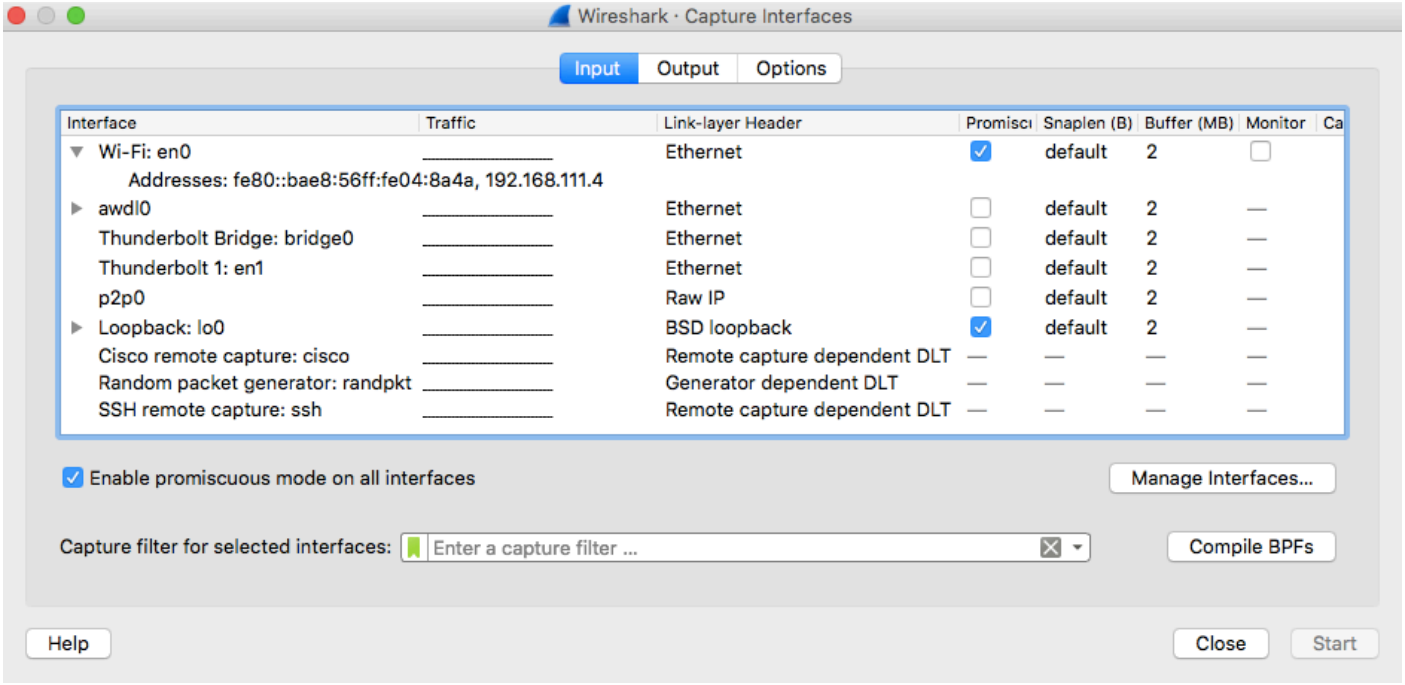


Рис. 5. Список сетевых интерфейсов

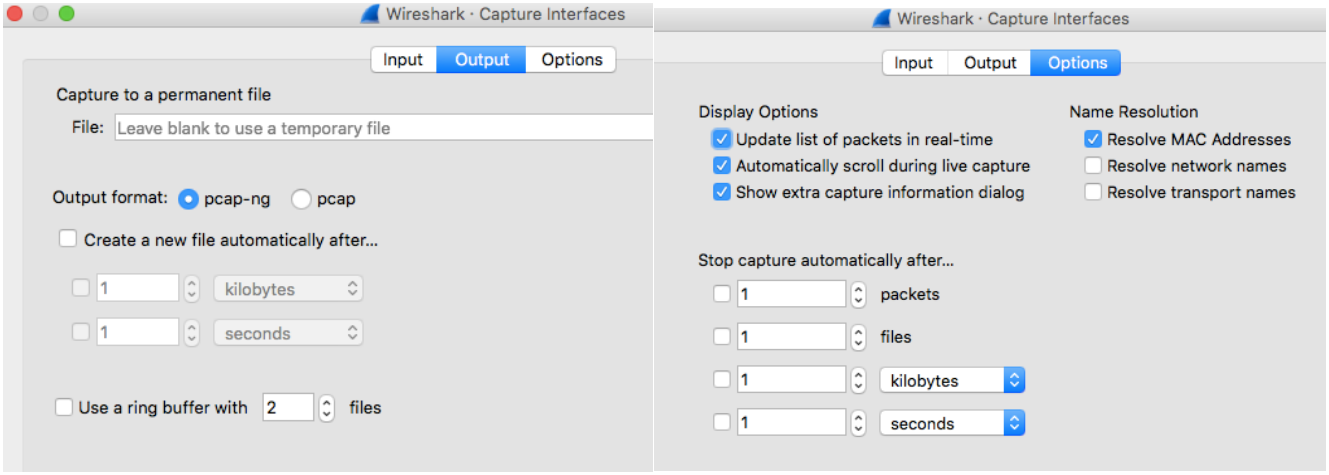


Рис. 6. Настройки перехвата

Для начала перехвата можно нажать кнопку Start на панели инструментов или воспользоваться меню Capture → Start.

После нажатия на кнопку Start начнется перехват пакетов. Если сетевая активность высокая, то можно будет сразу увидеть массу входящих и/или исходящих пакетов.

Для остановки перехвата необходимо нажать кнопку Stop или воспользоваться меню Capture → Stop.

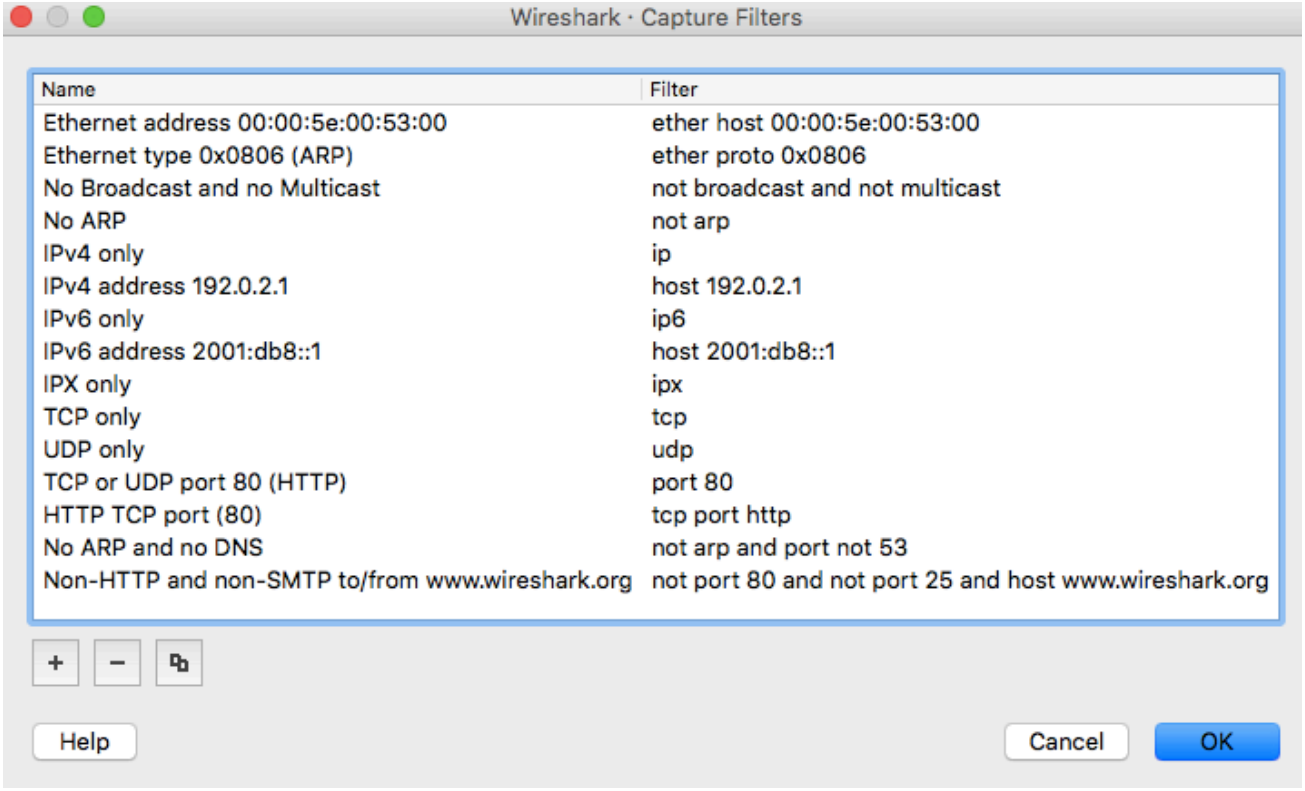


Рис. 7. Окно Capture Filters

4.4.2. Analyze and Statistics.

Кроме того, в Wireshark есть несколько удобных и полезных функций. Например, Analyse → Expert Information покажет список (рис. 8) основных событий, которые произошли во время захвата — открытие новых сессий, не совсем хорошее поведение протоколов (повторные квитанции в TCP, повторные передачи сегментов и т.д.).

Analyse →Follow (TCP|UDP|SSL) Stream — позволяет собрать сессию передачи воедино и посмотреть ее содержимое в целом - вплоть до восстановления переданной в течение сессии HTML-страницы.

Statistics → Capture File Properties позволяет просмотреть некоторую статистику в целом по сессии захвата — в том числе, среднее количество пакетов в секунду и объем передаваемых данных (рис. 9).

Statistics → Protocol Hierarhy — статистику по используемым протоко- лам, в том числе — в процентном соотношении (рис. 10).

Statistics → Conversations показывает информацию об участниках связи, кто кому сколько передавал пакетов, данных и в какую сторону (рис. 11).

Statistics → IO Graphs позволяет построить почти произвольный статистический график по захваченным данным (рис. 12).

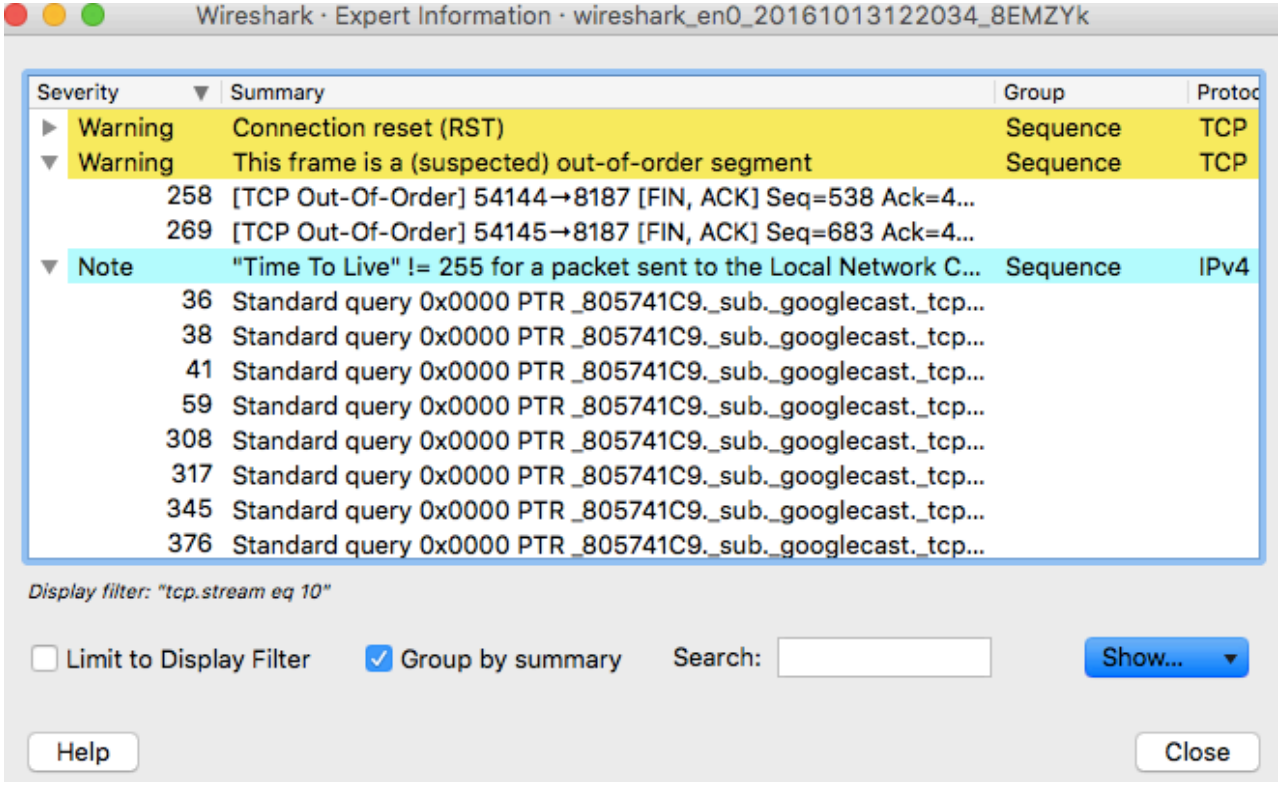


Рис. 8. Окно Expert Information

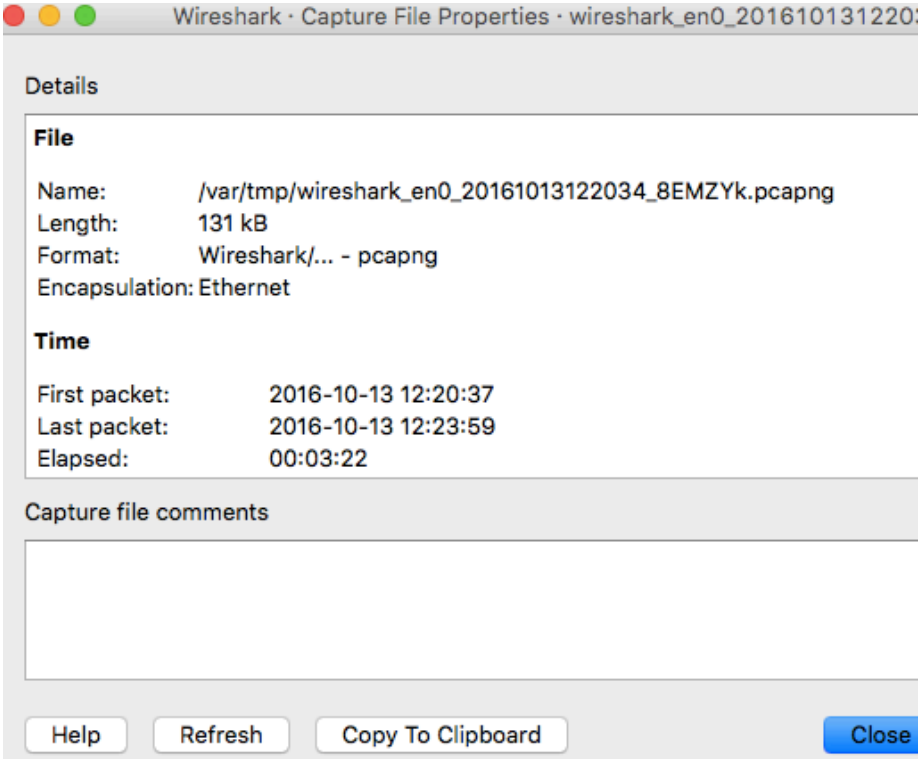
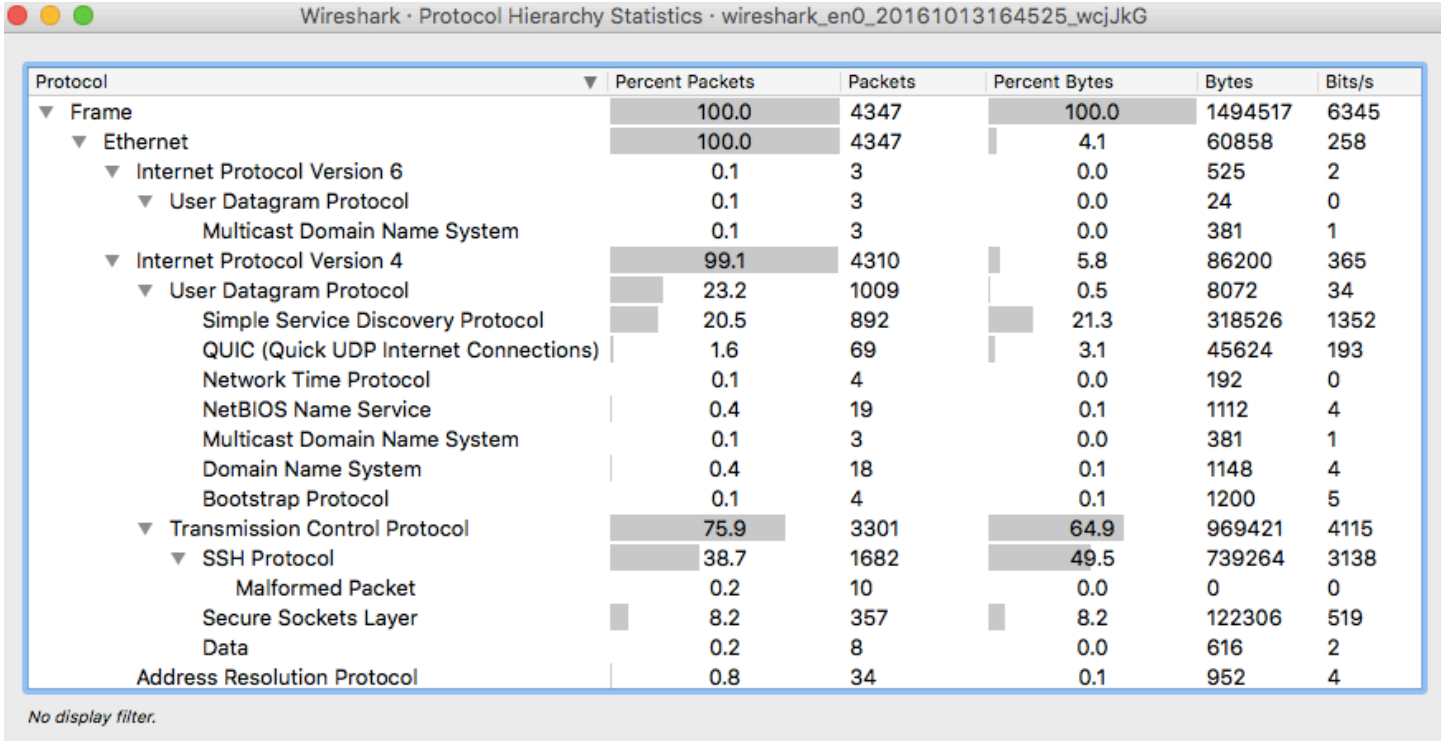


Рис. 9. Окно Capture File Properties



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s
Frame	100.0	4347	100.0	1494517	6345
Ethernet	100.0	4347	4.1	60858	258
Internet Protocol Version 6	0.1	3	0.0	525	2
User Datagram Protocol	0.1	3	0.0	24	0
Multicast Domain Name System	0.1	3	0.0	381	1
Internet Protocol Version 4	99.1	4310	5.8	86200	365
User Datagram Protocol	23.2	1009	0.5	8072	34
Simple Service Discovery Protocol	20.5	892	21.3	318526	1352
QUIC (Quick UDP Internet Connections)	1.6	69	3.1	45624	193
Network Time Protocol	0.1	4	0.0	192	0
NetBIOS Name Service	0.4	19	0.1	1112	4
Multicast Domain Name System	0.1	3	0.0	381	1
Domain Name System	0.4	18	0.1	1148	4
Bootstrap Protocol	0.1	4	0.1	1200	5
Transmission Control Protocol	75.9	3301	64.9	969421	4115
SSH Protocol	38.7	1682	49.5	739264	3138
Malformed Packet	0.2	10	0.0	0	0
Secure Sockets Layer	8.2	357	8.2	122306	519
Data	0.2	8	0.0	616	2
Address Resolution Protocol	0.8	34	0.1	952	4

Рис. 10. Окно Protocol Hierarchy

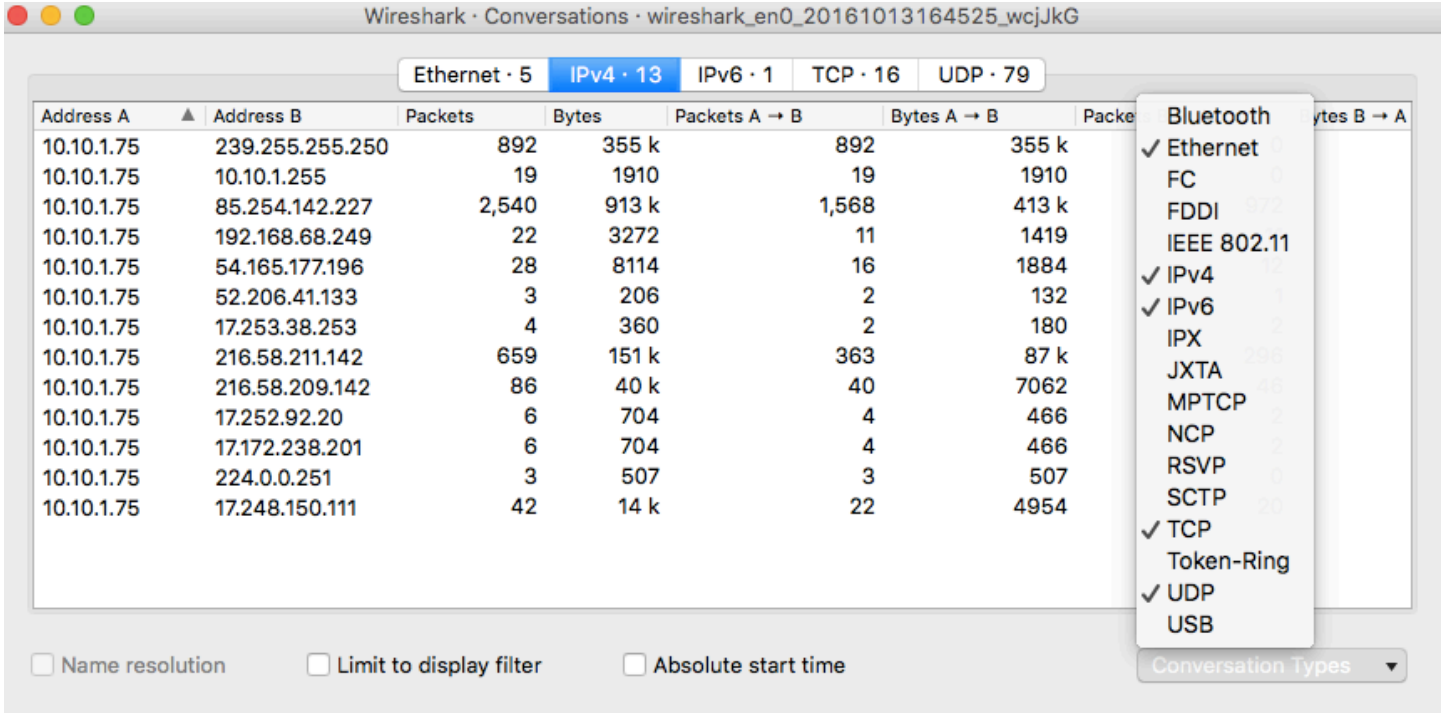


Рис. 11. Окно Conversations

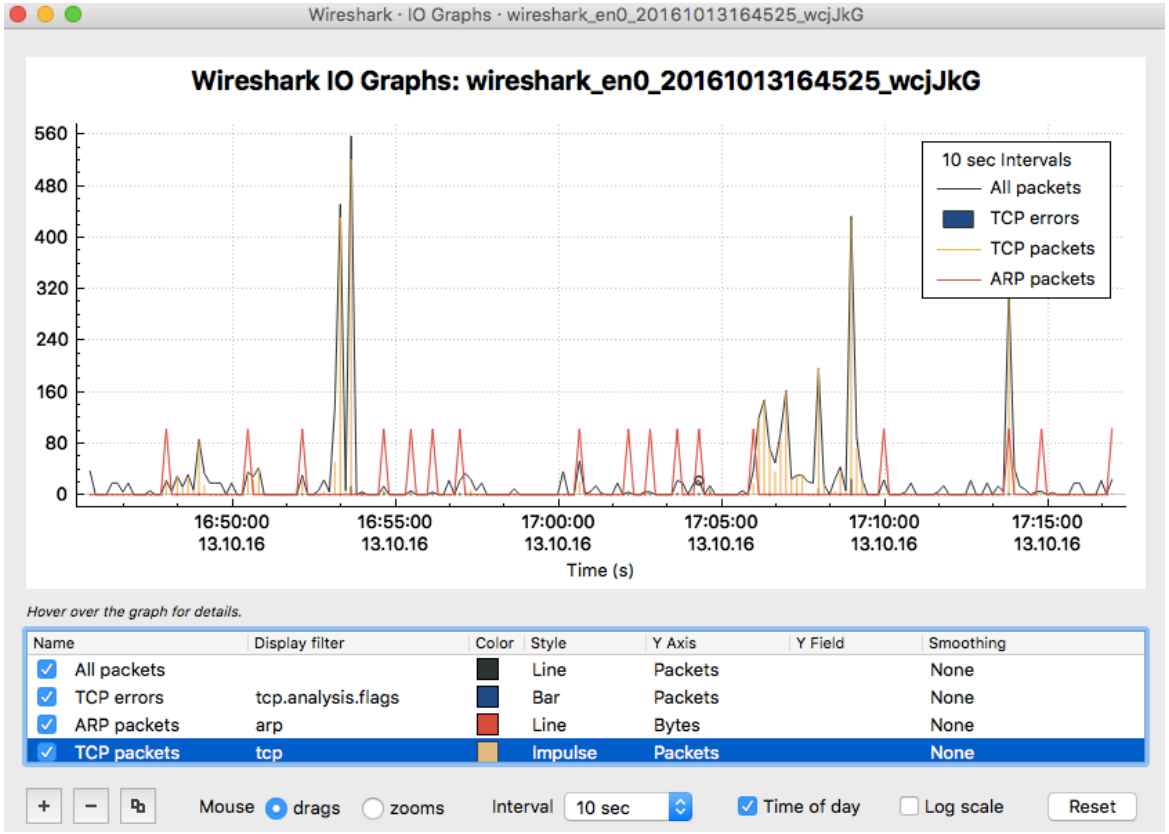


Рис. 12. Окно IO Graphs

5. Задание на лабораторную работу и отчёт.

5.1. Цель работы: Освоить базовые навыки мониторинга сети с использованием программ для анализа протоколов Wireshark и Network Miner.

5.2. Задание для лабораторной работы.

1. Загрузка и установка Wireshark.

Загрузите и установите Wireshark с сайта http://www.wireshark.org/download.html

Если у вас есть проблемы при загрузке или установке, то проконсультируйтесь на сайте http://wiki.wireshark.org/CaptureSetup

Мы рекомендуем вам использовать **Wireshark Portable Version** с сайта: http://net.academy.lv/soft/WiresharkPortable.zip

2. Загрузка и открытие файла с примером сбора трафика.

Загрузить в Wireshark .pcap файл с ранее собранным сетевым трафиком: из архива http://net.academy.lv/soft/pcap.zip взять файл SmallFlow.pcap.

3. Отчет по лабораторной работе.

Заполнить таблицы. Приложите ScreenShots. Отправьте отчет на e-mail.

5.3. LAB WORK REPORT

Student Name:	Student ID:	Date:

3.1. Capture File Properties

Заполнить таблицу. Исходные данные доступны в Statistics/Capture File Properties.

Nr	Parametr	Value
1	Time of capture, min	
2	Packets	
3	Bytes, MiB	
4	Average packet size, B	
5	Average packets per seconds, pps	
6	Average bytes per seconds, B/s	

Определить относительную загрузку сети L (в %) за контрольный период времени T по формуле:
L = (Traffic [Mbits] / T [sec]) / (Bandwidth [Mbits/sec])
Обратите внимание на соблюдение размерности (b-bit, B-byte, Kib, MiB, Mib)!

3.2. Ethernet Traffic Distribution by Protocols

Заполнить таблицу распределения трафика по протоколам IP, TCP, UDP, ICMP, ARP,...
Исходные данные доступны в Statistics/Protocol Hierarchy.

Nr	Protocol	Traffic, MiB	Traffic, %
1	IPv6		
2	IPv4		
3	--UDP		
4	--TCP		
5	--ICMP		
6	ARP		
7	802.1X		
	SUMM		**100**

Каково количественное соотношение прикладных и служебных протоколов?

3.3. Ethernet Traffic Distribution by Nodes

Составить таблицу распределения Ethernet-трафика по узлам сети (для 5 наиболее активных сетевых узлов по количеству Bytes). Исходные данные доступны в Statistics/Endpoints/Ethernet.

Nr	MAC-address	IP- address	Traffic					
			Rx input		Tx output		overall	
			MiB	%	MiB	%	MiB	%
1.								
2.								
3.								
4.								
5.								
		SUM		**100**		**100**		**100**

Какие из узлов являются наиболее загруженными с учетом направления трафика (исходящий, входящий, общий)?

3.4. Display Filters

Заполнить таблицу. Написать и проверить в Wireshark пять простых фильтров поиска с использованием И, ИЛИ, НЕТ для показа пакетов от(к) определённого(му) узла(у) формируемых ICMP, DNS, ARP запросами (ответами) при обращении к какому-либо серверу на ваш выбор.

Nr	Display Filter	Description
1		
2		
3		
4		
5		

3.5. Network Problem Analyse

Анализ пяти проблем в сети. Исходные данные доступны в Analyze/Expert Information.

Nr	Expert Information	Severity	Problem Description
1	Connection reset (RST)	Warning	
2	TCP keep-alive segment	Note	
3	...	Error	
4			
5			

5.4. EXTEND LAB WORK ASSIGNMENT

Задание для домашней работы.

1. Познакомиться с возможностями сетевого анализатор Network Miner. Загрузить в Network Miner предложенный преподавателем .pcap файл с ранее собранным сетевым трафиком. Проанализируйте собранный трафик в Network Miner.
2. Установить Wireshark на домашнем компьютере.
3. Запустить Wireshark в режиме захвата трафика, проходящего через интерфейс, подключенный к локальной сети (обычно это eth0).
4. Эмулировать сетевую активность в течении 10-15 минут с различных домашних узлов. Для этого можно выполнить, например, некоторые из указанных действий.
    - Открыть сайт http://….;
    - Подключиться к серверу ftp;
    - Подключиться к серверу mail;
    - Подключиться к серверу ssh;
    - Выполнить пинг любых узлов;
    - Подключиться к одному из доступных сетевых дисков Windows (если есть в сети);
    - Выполнить прочие действия, требующие сетевого подключения.
5. Остановить захват, сохранить pcap файл и приложить его к отчёту (если файл больше 10 MiB, то иметь его на флэшке при защите лабораторной работы.
6. Остальные пункты такие же как с 1 по 5 для Core Lab Work.
7. Предоставьте отчёт в электронной форме.