Лабораторная работа №4. Изучение работы программ-снифферов

Цель работы: изучить основы работы программ-снифферов. При помощи программы-сниффера WareShark осуществить перехват трафика и выполнить его анализ.

Задачи, решаемые с помощью снифферов

В работе компьютерной сети и сетевого стека узлов иногда возникают проблемы, причины которых трудно обнаружить общеизвестными утилитами сбора статистики (такими, например, как netstat) и стандартными приложениями на основе протокола ICMP (ping, traceroute/tracert и т.п.). В подобных случаях для диагностики неполадок часто приходится использовать более специфичные средства, которые позволяют отобразить (прослушать) сетевой трафик и проанализировать его на уровне единиц передачи отдельных протоколов («сниффинг», sniffing).

Анализаторы сетевых протоколов или «снифферы» являются исключительно полезными инструментами для исследования поведения сетевых узлов и выявления неполадок в работе сети. Разумеется, как и всякое средство, например, острый нож, сниффер может быть, как благом в руках системного администратора или инженера по информационной безопасности, так и орудием преступления в руках компьютерного злоумышленника [1].

Целью атак с использованием сниффера является похищение информации, обычно такой, как идентификационные номера пользователей, данные о функционировании сети, номера кредитных карт и.т.д.

Подобное специализированное программное обеспечение обычно использует «беспорядочный» (promiscuos) режим работы сетевого адаптера компьютера-монитора (в частности, для перехвата трафика сетевого сегмента, порта коммутатора или маршрутизатора). Как известно, суть данного режима сводится к обработке всех приходящих на интерфейс фреймов, а не

только предназначенных MAC-адресу сетевой карты и широковещательных, как это происходит в обычном режиме.

Применение снифферов на разных сетевых уровнях

Важно помнить о том, что атаки с использованием сниффера могут затрагивать диапазон от 1 до 7 уровня OSI. Если говорить о физическом соединении, кто-либо, уже имеющий доступ к внутренней локальной сети (чаще всего это работник компании), может использовать программы для прямого захвата сетевого трафика. Применяя техники спуфинга, взломщик, находящийся за пределами атакуемой сети, может вести перехват пакетов на уровне межсетевого экрана и похищать данные. В последнее время все чаще используется атака, направленная на перехват данных беспроводных сетей, при которой задача атакующего упрощается, так как нужно всего лишь находиться в радиусе действия сети для сбора информации о ней и проникновения в нее.

Использование сниффера в сетях, работающих по протоколу TCP/IP подразумевает захват, декодирование, исследование и интерпретацию данных, передающихся в пакетах по сети.

Для понимания того, для чего взломщики используют снифферы, нам следует знать о том, какие данные они могут получить из сети. Таблица 1.1 иллюстрирует уровни OSI и ту информацию, которой взломщик может завладеть на каждом уровне, успешно использовав сниффер.

Таблица 1.1. Распределение уровней OSI

Уровень	Действие
Прикладной	Захват имен пользователей и паролей
Представительский	Захват трафика сессий SSL/TLS
Сеансовый	Захват трафика Telnet/FTP
Транспортный	Захват TCP-сессий и UDP-трафика
Сетевой	Захват IP-адресов и номеров портов
Канальный	Захват MAC-адресов и ARP-запросов
Физический	Получение сведений о сети

Пакет TCP/IP содержит информацию, необходимую для соединения двух сетевых интерфейсов. Он содержит такие поля с информацией об исходном и целевом IP-адресах, номерах портов, номере пакета и типе протокола. Каждое из этих полей является необходимым для функционирования различных уровней сетевого стека и особенно приложений, относящихся к прикладному уровню (уровню 7 OSI), обрабатывающих принятые данные.

По своей природе протокол TCP/IP занимается только тем, что проверяет, сформирован ли пакет, добавлен ли он в Ethernet-фрейм и доставлен ли он от отправителя по сети к адресату. Однако, в этом протоколе не имеется механизмов для контроля безопасности данных. Таким образом, задача по установлению того, не произошло ли вмешательство в передачу данных, перекладывается на высшие уровни сетевого стека.

Способы перехвата траффика

В зависимости от того, каково нахождение взломщиков в сети, где производится перехват данных, они используют программы для захвата или программы для исследования пакетов.

В отношении технической стороны захвата пакетов следует помнить о том, что программы, осуществляющие захват пакетов, всегда работают в promiscous-режиме, что делает возможным захват и сохранение всех данных, передающихся по сети, с их помощью. Это также означает то, что даже если пакет не предназначается для сетевого интерфейса, на котором работает сниффер, он все равно будет захвачен, сохранен и проанализирован.

В ходе атак, заключающихся в захвате пакетов, используются снифферы, являющиеся либо программным обеспечением с открытым исходным кодом, либо коммерческим программным обеспечением. В целом, существуют три пути перехвата трафика в сети:

– использование внешней сети для перехвата траффика;

– использование внутренней сети для перехвата траффика;

– использование беспроводной сети для перехвата траффика.

Похищение паролей (Web password sniffing)

Как становится ясно из названия, в ходе атаки перехватываются данные HTTP-сессий и из них выделяются идентификаторы пользователей и пароли, которые похищаются. Хотя для защиты HTTP-сессий от таких атак и разработан протокол SSL, существует множество сайтов во внутренних сетях, использующих стандартное менее безопасное шифрование. Достаточно просто перехватить данные зашифрованные по алгоритмам Base64 или Base128 и получить пароль, применив специальное программное обеспечение. В современных снифферах присутствует функция захвата и получения информации, передаваемой в рамках SSL-сессий, но использовать эту функцию непросто.

Похищение TCP-сессий (TCP session stealing)

Этот метод заключается в простом захвате трафика между IP-адресом отправителя и адресата, проходящего через сетевой интерфейс в promiscous-режиме. Такие подробности, как номера портов, типы служб, порядковые номера TCP и сами данные интересуют взломщиков в первую очередь. После захвата достаточного количества пакетов, опытные взломщики могут самостоятельно создавать TCP-сессии, вводя в заблуждение узлы, являющиеся источником и адресатом пакетов, а также осуществлять атаку перехвата с участием человека (man-in-the-middle) в отношении активной TCP-сессии.

Захват данных приложений (Application-level sniffing)

Обычно из захваченных пакетов данных можно получить некоторое количество информации относительно приложений, осуществляющих обмен данными, и на основе этой информации провести другие атаки или просто похитить эту информацию. Например, протокол захвата данных может быть исследован с целью идентификации операционной системы, анализа SQL-за-

просов, получения информации о TCP-портах, специфических для приложения и т. д. С другой стороны, создание списка приложений, исполняющихся на сервере является хорошим началом атаки в отношении этих приложений.

Захват данных в локальной сети (LAN sniff)

Сниффер, работающий во внутренней сети может захватывать данные со всего диапазона IP-адресов. Это помогает злоумышленнику получить данные о функционировании сети, такие, как список активных узлов, список открытых портов, данные об оборудовании серверов и другие. Как только получен список открытых портов, становится возможной атака на основе эксплуатации уязвимостей отдельных служб, работающих на определенных портах [11].

Захват информации об используемых протоколах (Protocol sniff)

Этот метод подразумевает захват данных, относящихся к различным протоколам, используемым в сети. Сначала создается список протоколов, на основе захваченных данных. Этот список в будущем может использоваться для захвата данных, относящихся к отдельным протоколам. Например, если данных, относящихся к протоколу ICMP не было обнаружено во время захвата, считается, что этот протокол заблокирован. Однако, если при захвате обнаружены UDP-пакеты, отдельный сниффер для UDP-трафика начинает использоваться для захвата и расшифровки трафика, относящегося к Telnet, PPP, DNS и другим приложениям.

Захват ARP-трафика (ARP sniff)

В ходе этого популярного метода атаки злоумышленник захватывает как можно больший объем данных для создания таблицы соответствия IP-адресов MAC-адресам. Эта таблица впоследствии может быть использована для подмены ARP-записей (APR poisoning), спуфинг-атак или для эксплуатации уязвимостей маршрутизатора [10].

Роль снифферов

Использование сниффера считается типом "пассивной" атаки, при котором атакующие не могут быть замечены в сети. Это обстоятельство затрудняет определение наличия данной атаки и, поэтому, этот тип атаки является опасным.

Использование сниффера помогает взломщикам либо получить информацию непосредственно из сетевого трафика, либо получить данные о работе сети, которые могут быть использованы для подготовки последующих атак. Взломщики очень часто прибегают к использованию сниффера, так как возможно длительное его использование без риска быть разоблаченным.

Современные снифферы предназначаются для диагностики сетей, но при этом также могут быть использованы и для взлома. В таблице 1.2 приведены основные примеры, описывающую законные и незаконные примеры использования снифферов.

Таблица 1.2. Цели использования снифферов

Примеры законного использования	Примеры незаконного использования
1. Захват пакетов. 2. Анализ использования трафика в сети. 3. Преобразования пакетов для анализа данных. 4. Диагностика сетей.	1. Похищение пользовательских паролей и идентификаторов. 2. Похищение данных, относящихся к сообщениям электронной почты и служб мгновенных сообщений. 3. Похищение данных с применением спуфинга. 4. Похищение средств и причинение ущерба репутации.

Сниффер Wireshark

Утилита Wireshark является широко известным инструментом перехвата и интерактивного анализа сетевого трафика, фактически, стандартом в промышленности и образовании. К ключевым особенностям Wireshark можно отнести: многоплатформенность (Windows, Linux, Mac OS, FreeBSD,

Solaris и др.); возможности анализа сотен различных протоколов; поддержку как графического режима работы, так и интерфейса командной строки (утилита tshark); мощную систему фильтров трафика; экспорт результатов работы в форматы XML, PostScript, CSV и т. д.

Немаловажным фактом является также то, что Wireshark — это программное обеспечение с открытым исходным кодом, распространяемое под лицензией GNU GPLv2, т. е. Вы можете свободно использовать этот продукт по своему усмотрению.

Установка Wireshark

Последнюю версию Wireshark для операционных систем Windows и OS X, а также исходный код можно скачать с сайта проекта. Для дистрибутивов Linux и BSD-систем, данный продукт обычно доступен в стандартных или дополнительных репозиториях. Расматриваемые в данном издании снимки экранов сделаны с версии 1.10.10 Wireshark для Windows. Более ранние версии программы, которые можно найти в репозиториях Unix-подобных операционных систем, также можно успешно использовать, так как Wireshark давно уже стабильный и функциональный продукт.

Работа Wireshark базируется на библиотеке Pcap (Packet Capture), предоставляющей собой прикладной интерфейс программирования для реализации низкоуровневых функций взаимодействия с сетевыми интерфейсами (в частности перехвата и генерации произвольных единиц передачи сетевых протоколов и протоколов локальных сетей). Библиотека Pcap является также основой таких известных сетевых средств, как tcpdump, snort, nmap, kismet и т. д. Для Unix-подобных систем Pcap обычно присутствует в стандартных репозиториях программного обеспечения. Для семейства операционных систем Windows существует версия Pcap, которая называется Winpcap. Ее можно скачать с сайта проекта. Впрочем, обычно в этом нет

необходимости, так как библиотека Winpcap включена в пакет установки Wireshark для Windows.

Процесс установки программы не сложен для любой операционной системы, с поправкой, разумеется, на специфику используемой Вами платформы. Например, Wireshark в Debian/Ubuntu устанавливается так, что непривилегированные пользователи по умолчанию не имеют права перехватывать пакеты, поэтому программу нужно запускать с использованием механизма смены идентификатора пользователя sudo (или же произвести необходимые манипуляции согласно документации стандартного DEB-пакета).

Основы работы с Wireshark

Пользовательский интерфейс Wireshark (рисунок 1.1.) построен на основе библиотеки GTK+ (GIMP Toolkit). Главное окно программы включает следующие элементы: меню, панели инструментов и фильтров просмотра, список пакетов, детальное описание выбранного пакета, отображение байтов пакета (в шестнадцатеричной форме и в виде текста) и строку состояния:

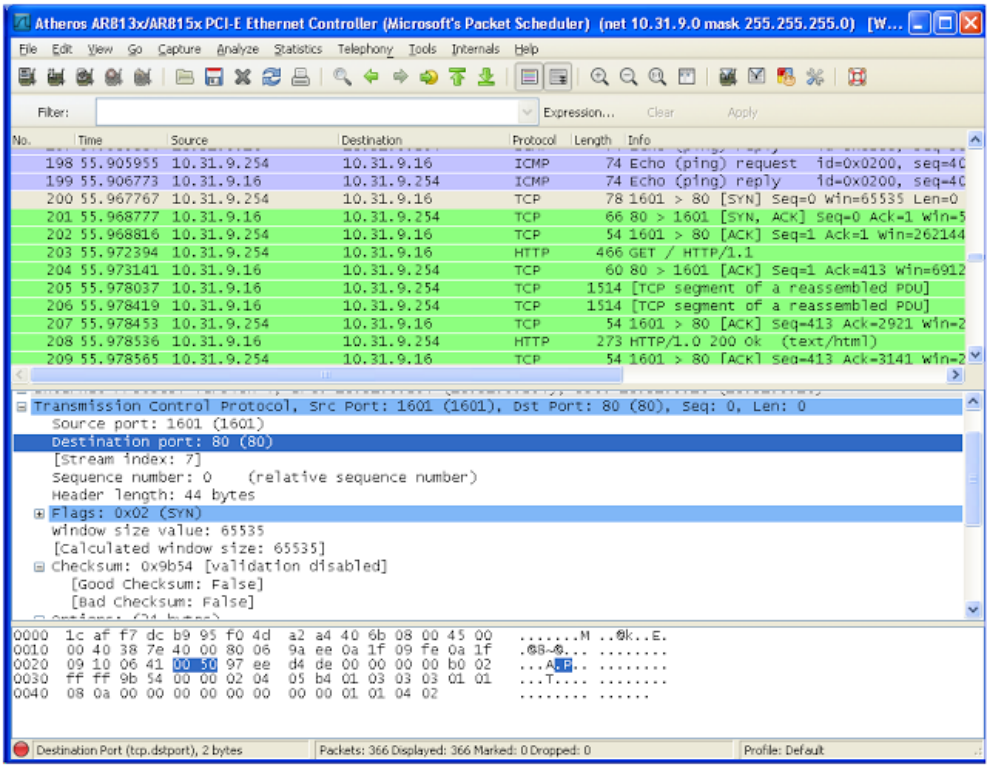

Рисунок 1.1. Основное окно программы Wireshark

Следует отметить, что пользовательский интерфейс программы хорошо проработан, достаточно эргономичен и вполне интуитивен, что позволяет пользователю сконцентрироваться на изучении сетевых процессов, не отвлекаясь по мелочам. Кроме того, все возможности и подробности использования Wireshark детально описаны в руководстве пользователя. Поэтому мы уделим основное внимание функциональным возможностям рассматриваемого продукта.

Итак, эргономика Wireshark отражает многоуровневый подход к обеспечению сетевых взаимодействий. Все сделано таким образом, что, выбрав сетевой пакет из списка, пользователь получает возможность просмотреть все заголовки (слои), а также значения полей каждого слоя сетевого пакета, начиная от обертки — кадра Ethernet, непосредственно IP-заголовка, заголовка транспортного уровня и данных прикладного протокола, содержащихся в пакете.

Исходные данные для обработки могут быть получены Wireshark в режиме реального времени или импортированы из файла дампа сетевого трафика, причем несколько дампов для задач анализа можно «на лету» объединить в один.

Проблема поиска необходимых пакетов в больших объемах перехваченного трафика решается двумя типами фильтров: сбора трафика (capture filters) и его отображения (display filters). Фильтры сбора Wireshark основаны на языке фильтров библиотеки Pcap, т.е. синтаксис в данном случае аналогичен синтаксису утилиты tcpdump. Фильтр представляет собой серию примитивов, объединенных, если это необходимо, логическими функциями (and, or, not). Часто используемые фильтры можно сохранять в профиле для повторного использования.

На рисунке 1.2. показан профиль фильтров сбора Wireshark:

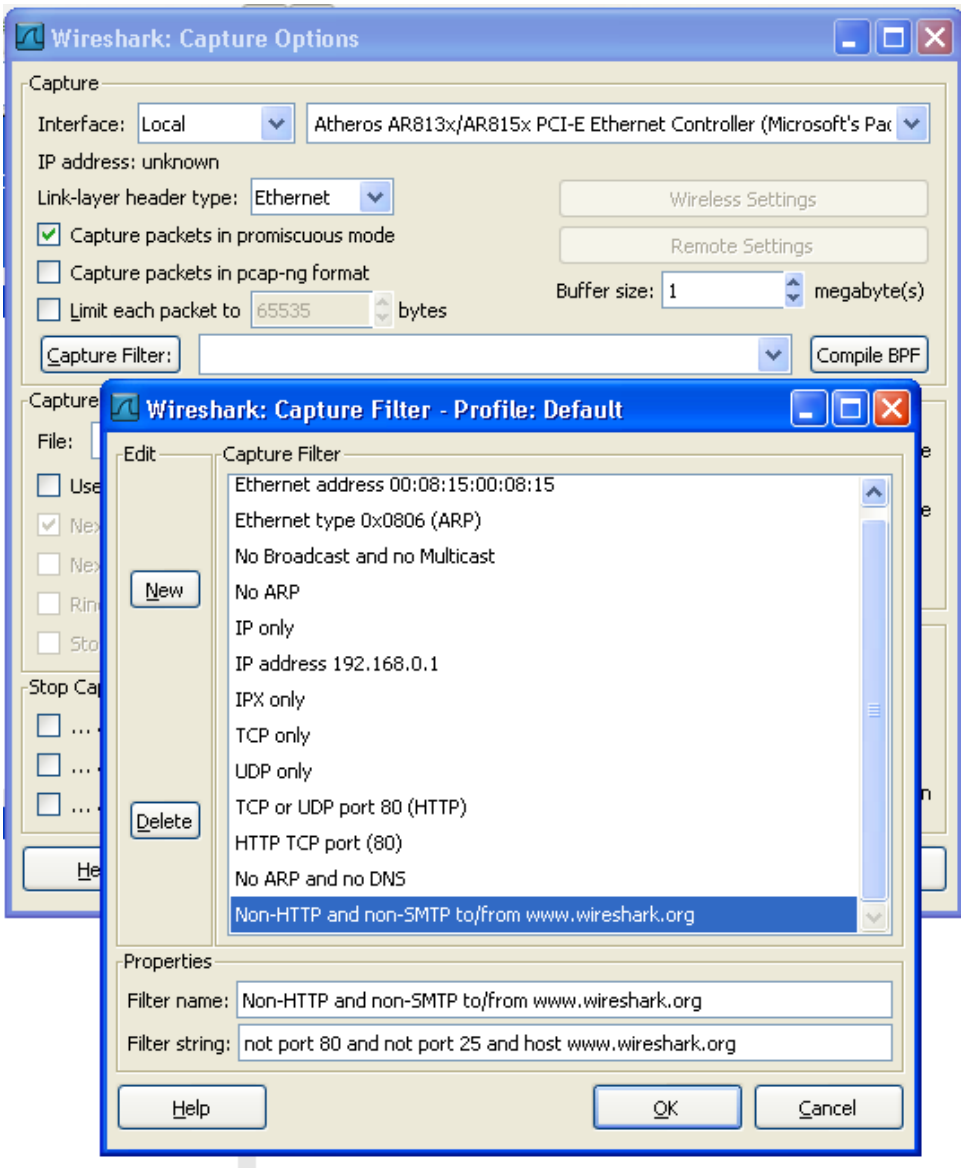

Рисунок 1.2. Работа с профилями фильтров в Wireshark

Анализатор сетевых пакетов Wireshark также имеет свой простой, но многофункциональный язык фильтров отображения. Значение каждого поля в заголовке пакета может быть использовано как критерий фильтрации (например, ip.src — IP-адрес источника в сетевом пакете, frame.len — длина Ethernet-фрейма и т.д.). С помощью операций сравнения значения полей можно сопоставлять заданным величинам (например, frame.len <10), а несколько выражений объединять логическими операторами (например,

ip.src==10.0.0.5 and tcp.flags.fin). Хорошим помощником в процессе конструирования выражений является окно настройки правил отображения (Filter Expression) – рисунок 1.3:

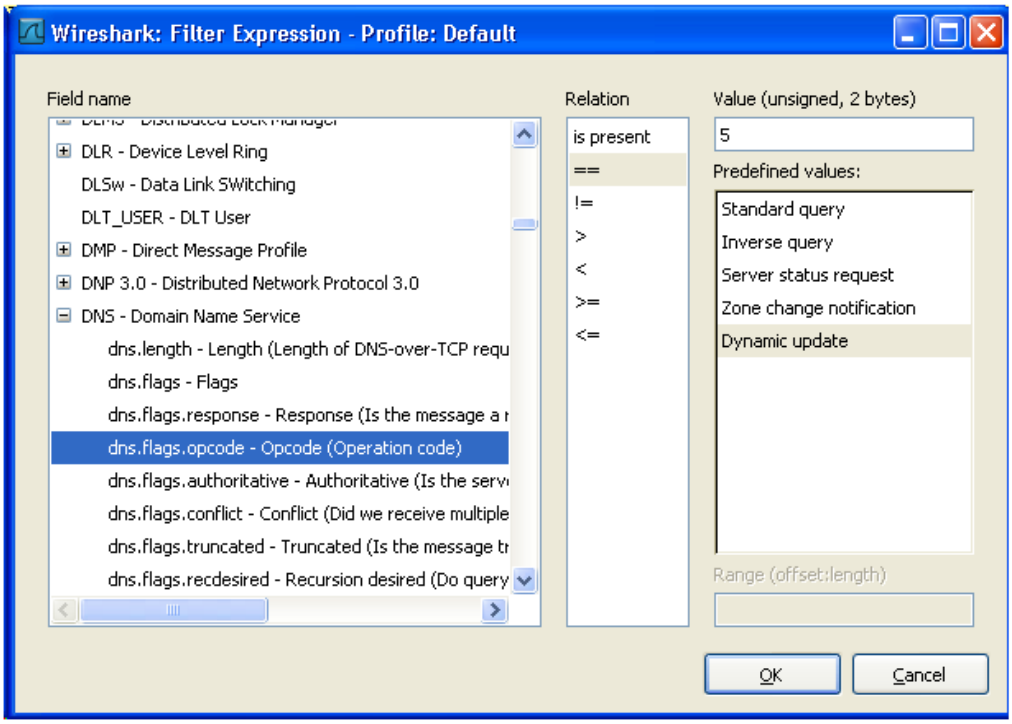

Рисунок 1.3. Конструирование выражений фильтра в утилите Wireshark

Средства анализа сетевых пакетов

Если протоколы без установления соединения возможно исследовать простым просмотром отдельных пакетов и расчетом статистики, то изучение работы ориентированных на соединение протоколов существенно упрощается при наличии дополнительных возможностей анализа хода сетевых взаимодействий.

Одной из полезных функций Wireshark является пункт «Follow TCP Stream» (буквально, «Следовать за TCP-потоком») подменю анализа «Analyze», позволяющий извлечь данные прикладного протокола из TCP-сегментов потока, которому принадлежит выбранный пакет рисунок 1.4:

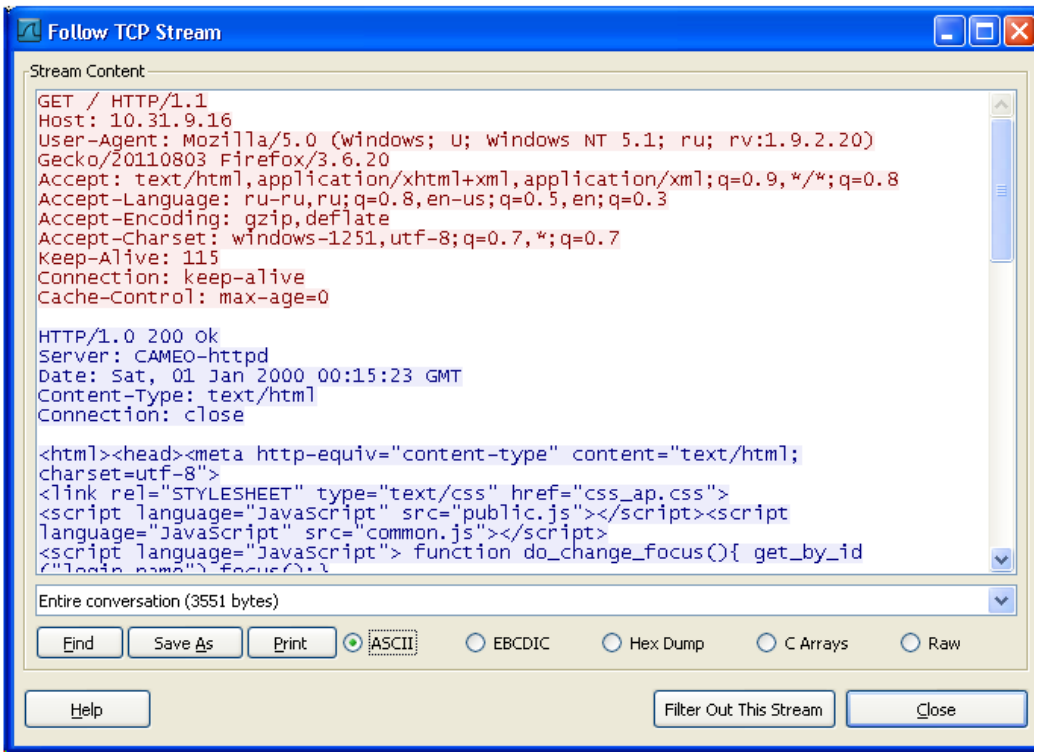

Рисунок 1.4. Извлечение данных из TCP потока в Wireshark

Еще один интересный пункт подменю анализа – «Expert Info Composite», вызывающий окно встроенной экспертной системы Wireshark, которая попытается обнаружить ошибки и замечания в пакетах, автоматически выделить из дампа отдельные соединения и охарактеризовать их. Данный модуль находится в процессе разработки и совершенствуется от версии к версии программы.

В подменю статистики «Statistics» собраны опции, позволяющие рассчитать всевозможные статистические характеристики изучаемого трафика, построить графики интенсивности сетевых потоков, проанализировать время отклика сервисов и т.д. Так, пункт «Protocol Hierarchy» отображает статистику в виде иерархического списка протоколов с указанием процентного отношения к общему трафику, количества пакетов и байт, переданных данным протоколом.

Функция «Endpoint» дает многоуровневую статистику по входящему/исходящему трафику каждого узла. Пункт «Conversations» (буквально,

«разговоры») позволяет определить объемы трафика различных протоколов (канального, сетевого и транспортного уровня модели взаимодействия открытых систем), переданного между взаимодействовавшими друг с другом узлами. Функция «Packet Lengths» отображает распределение пакетов по их длине.

Пункт «Flow Graph...» представляет потоки пакетов в графическом виде. При этом, при выборе элемента на графике становится активным соответствующий пакет в списке в главном окне программы – рисунок 1.5.

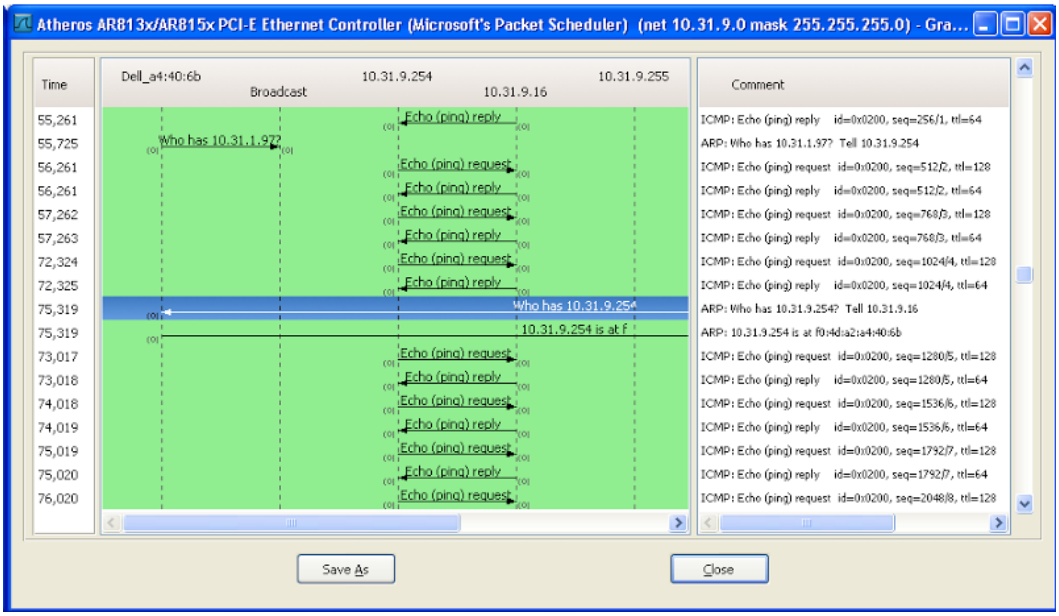

Рисунок 1.5. Графическая визуализация пакетов данных в Wireshark

Отдельное подменю в последних версиях Wireshark отведено IP-телефонии. В подменю «Tools» есть пункт «Firewall ACL Rules», для выбранного пакета попытается создать правило межсетевого экрана (в версии 1.6.x поддерживаются форматы Cisco IOS, IP Filter, IPFirewall, Netfilter, Packet Filter и Windows Firewall).

Программа также имеет встроенный интерпретатор легковесного языка программирования Lua. Используя Lua, Вы можете создавать собственные «декодеры» протоколов и обработчики событий в Wireshark.

Анализатор сетевых пакетов Wireshark является примером Opensource-продукта, успешного как в рамках платформы Unix/Linux, так популярного среди пользователей Windows и Mac OS X.

Существуют консольные варианты снифферов. Для Windows используется утилита WinDump или TShark, для UNIX-систем – tcpdump [11].

Задания:

1. Освоить интерфейс работы с программой Wireshark. Изучить основные принципы ее работы. Осуществить захват пакетов с HTTP трафиком и провести его анализ.

2. Освоить интерфейс работы с программой Wireshark. Изучить основные принципы ее работы. Осуществить захват пакетов с FTP трафиком и провести его анализ.

3. Освоить интерфейс работы с программой Wireshark. Изучить основные принципы ее работы. Осуществить захват пакетов с SMTP трафиком и провести его анализ.

4. Освоить интерфейс работы с программой Wireshark. Изучить основные принципы ее работы. Осуществить захват пакетов с POP3 трафиком и провести его анализ.

5. Освоить интерфейс работы с программой Wireshark. Изучить основные принципы ее работы. Осуществить захват пакетов приложения, работающего с базой Microsoft SQL и провести его анализ.

6. Освоить интерфейс работы с программой Wireshark. Изучить основные принципы ее работы. Осуществить захват пакетов приложения, работающего с базой MySQL и провести его анализ.

7. Освоить интерфейс работы с программой Wireshark. Изучить основные принципы ее работы. Осуществить захват пакетов приложения, работающего с базой PostgreSQL и провести его анализ.

8. Освоить интерфейс работы с программой Wireshark. Изучить основные принципы ее работы. Осуществить захват пакетов с любым TCP трафиком и провести его анализ.

9. Освоить интерфейс работы с программой Wireshark. Изучить основные принципы ее работы. Осуществить захват пакетов с любым UDP трафиком и провести его анализ.