

Министерство науки и высшего образования Российской Федерации

Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

Ю.С. Белов, А.Н. Молчанов

ПРИМЕНЕНИЕ АНАЛИЗА ТРАФИКА В БЕСПРОВОДНЫХ СЕТЯХ

Методические указания к выполнению лабораторной работы
по курсу «Беспроводные технологии передачи данных»

Калуга – 2019


УДК 004.71
ББК 32.972.5
Б435

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационных технологий».

Методические указания рассмотрены и одобрены:


- Кафедрой «Программного обеспечения ЭВМ, информационных технологий» (ИУ4-КФ) протокол № 51.4/6 от «20» февраля 2019 г.

Зав. кафедрой ИУ4-КФ

 к.т.н., доцент Ю.Е. Гагарин

- Методической комиссией факультета ИУ-КФ протокол № 9 от «04» 03 2019 г.


Председатель методической
комиссии факультета ИУ-КФ

 к.т.н., доцент М.Ю. Адкин

- Методической комиссией

КФ МГТУ им.Н.Э. Баумана протокол № 5 от «5» 03 2019 г.

Председатель методической комиссии
КФ МГТУ им.Н.Э. Баумана

 д.э.н., профессор О.Л. Перерва

Рецензент:

к.т.н., доцент кафедры ИУ3-КФ

 А.В. Финюшин

Авторы

к.ф.-м.н., доцент кафедры ИУ4-КФ
ст. преп. кафедры ИУ6-КФ

 Ю.С. Белов
 А.Н. Молчанов

Аннотация

Методические указания по выполнению лабораторной работы по курсу «Беспроводные технологии передачи данных» содержат описание механизмов обеспечения безопасности в беспроводных сетях и анализа трафика в беспроводных сетях.

Предназначены для студентов 4-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

© Калужский филиал МГТУ им. Н.Э. Баумана, 2019 г.
© Ю.С. Белов, А.Н. Молчанов, 2019 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ.....	5
КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ	6
ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ	24
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ	24
ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ.....	24
ОСНОВНАЯ ЛИТЕРАТУРА.....	25
ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА	25

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Беспроводные технологии передачи данных» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 4-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткое описание настроек безопасности в беспроводных сетях, анализа трафика в беспроводных сетях и задание на выполнение лабораторной работы.

Методические указания составлены для ознакомления студентов с возможностями оборудования для беспроводных локальных сетей. Для выполнения лабораторной работы студенту необходимы минимальные знания архитектуры ЭВМ, компьютерных сетей и технологии локальных вычислительных сетей.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является получение практических навыков применения методов анализа трафика в беспроводных сетях.

Основными задачами выполнения лабораторной работы являются:

1. Ознакомиться с приложениями и методами, позволяющими осуществлять анализ трафика беспроводных сетей.
2. Научиться применять на практике механизмы сбора и анализа трафика.
3. Научиться реализовывать механизмы передачи данных с использованием стандарта IEEE 802.15.1.

Результатами работы являются:

- Собранная статистика по передаваемым данным
- Подготовленный отчет

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Термин снифер происходит от английского «to sniff» – нюхать – и представляет собой программу 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 (свитчей), в противном случае метод малоэффективен, поскольку на снифер попадают лишь отдельные фреймы);
- подключением снифера в разрыв канала;
- ответвлением (программным или аппаратным) трафика и направлением его копии на снифер;
- 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 снифер с последующим возвращением трафика в надлежащий адрес.

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

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

- обнаружить паразитный, вирусный и закольцованный трафик, наличие которого увеличивает загрузку сетевого оборудования и каналов связи (сниферы здесь малоэффективны; как правило, для

этих целей используют сбор разнообразной статистики серверами и активным сетевым оборудованием и её последующий анализ);

- выявить в сети вредоносное и несанкционированное ПО, например, сетевые сканеры, флудеры, троянские программы, клиенты пиринговых сетей и другие (это обычно делают при помощи специализированных sniffеров — мониторов сетевой активности);
- перехватить любой незашифрованный (а порой и зашифрованный) пользовательский трафик с целью получения паролей и другой информации;
- локализовать неисправность сети или ошибку конфигурации сетевых агентов (для этой цели sniffеры часто применяются системными администраторами)

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

Как ни странно, в природе существует великое множество sniffеров, поэтому их разделяют на категории:

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

Wireshark

Wireshark – это анализатор сетевого трафика. Его задача состоит в том, чтобы перехватывать сетевой трафик и отображать его в детальном виде. Анализатор сетевого трафика можно сравнить с измерительным устройством, которое используется для просмотра того, что происходит внутри сетевого кабеля, как например вольтметр используется электриками для того чтобы узнать, что происходит внутри электропроводки (но, конечно, на более высоком уровне). В прошлом такие инструменты были очень дорогостоящими и проприетарными. Однако, с момента появления такого инструмента как Wireshark ситуация изменилась. Wireshark – это один из лучших анализаторов сетевого трафика, доступных на сегодняшний момент. Wireshark работает на основе библиотеки pcap. Библиотека Pcap (Packet Capture) позволяет создавать программы анализа сетевых данных, поступающих на сетевую карту компьютера. Разнообразные программы мониторинга и тестирования сети, [сниферы](#) используют эту библиотеку. Она написана для использования языка C/C++ так что другие языки, такие как Java, .NET и скриптовые языки использовать не рационально. Для Unix-подобных систем используют libpcap библиотеку, а для Microsoft Windows NT используют WinPcap библиотеку.

Программное обеспечение сетевого мониторинга может использовать libpcap или WinPcap, чтобы захватить пакеты, путешествующие по сети и в более новых версиях для передачи пакетов в сети. Libpcap и WinPcap также поддерживают сохранение захваченных пакетов в файл и чтение файлов, содержащих сохранённые пакеты. Программы написанные на основе libpcap или WinPcap могут захватить сетевой трафик, анализировать его. Файл захваченного трафика сохраняется в формате, понятном для приложений, использующих Pcap.

Wireshark используется для:

- Системные администраторы используют его для решения проблем в сети.
- Аудиторы безопасности используют его для выявления проблем в сети.

- Разработчики используют его для отладки сетевых приложений.
- Обычные пользователи используют его для изучения внутреннего устройства сетевых протоколов.

Возможности Wireshark:

- Работает на большинстве современных ОС (Microsoft Windows, Mac OS X, UNIX). Wireshark – продукт с открытым исходным кодом, распространяемый на основании лицензии GPL. Его можно использовать на любом количестве компьютеров, не опасаясь за ввод лицензионных ключей, продление лицензии и другие неприятные мероприятия. Поэтому сообществу очень легко добавлять в него поддержку новых протоколов в виде плагинов или напрямую вшить её в исходный код.
- Перехват трафика сетевого интерфейса в режиме реального времени. Wireshark может перехватывать трафик различных сетевых устройств, отображая его имя (включая беспроводные устройства). Поддерживаемость того или иного устройства зависит от многих факторов, например, от операционной системы.
- Множество 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).
- Сохранение и открытие ранее сохраненного сетевого трафика.
- Импорт и экспорт файлов из других пакетных анализаторов. Wireshark может сохранять перехваченные пакеты в большое количество форматов других пакетных анализаторов, например: libpcap, tcpdump, Sun snoop, atmsnoop, Shomiti/Finisar Surveyor, Novell LANalyzer, Microsoft Network Monitor, AIX's iptrace.
- Позволяет фильтровать пакеты по множеству критерий.
- Позволяет искать пакеты по множеству критерий.
- Позволяет подсвечивать захваченные пакеты разных протоколов.
- Позволяет создавать разнообразную статистику.

Ниже перечислены некоторые вещи, которые Wireshark делать 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 резолвинге доменных имен, но и эту функцию можно отключить.

Установка

Наиболее стабильную версию программы (v1.0.6-1.0.7) можно скачать по адресу компании-разработчика программы <http://www.wireshark.org/download.html> .

Для установки программы необходимо запустить скачанный файл wireshark-setup-1.0.6.exe и в появившемся окне выбрать Next> (Рис. 1).

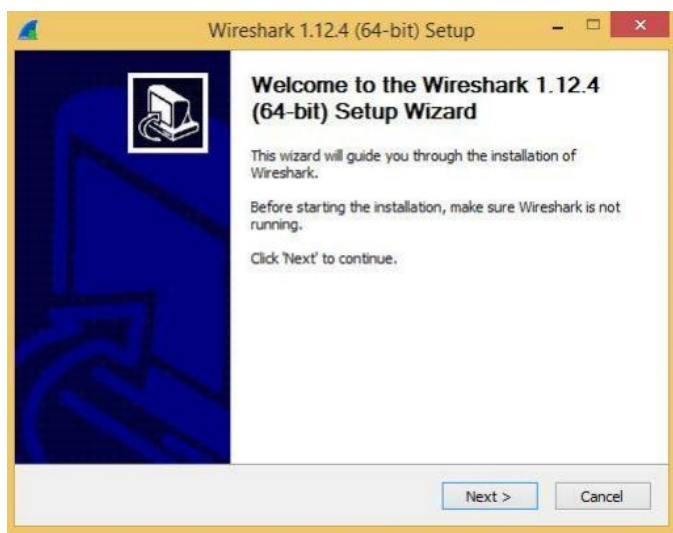


Рис. 1. Окно запуска установки

Далее Вам необходимо согласить с выбранными по умолчанию компонентами, предложенными для установки (Рис. 2):

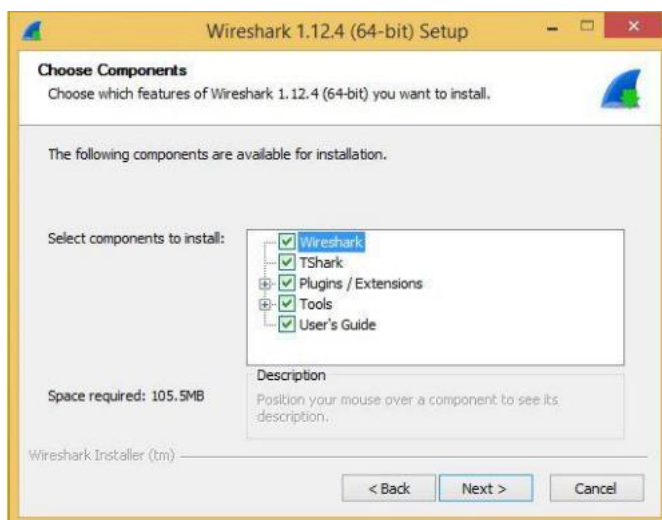


Рис. 2. Окно выбора компонентов установки

Далее соглашаемся с выбранными параметрами установки программы и выбираем путь для инсталляции (Рис. 3):

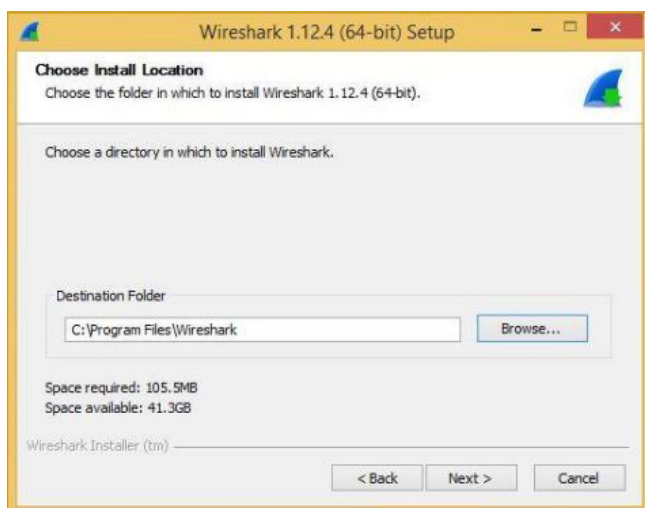


Рис. 3. Окно выбора пути установки

Процесс установки начался, далее инсталлятор попросит установить WinPcap – соглашаемся (Рис. 4):

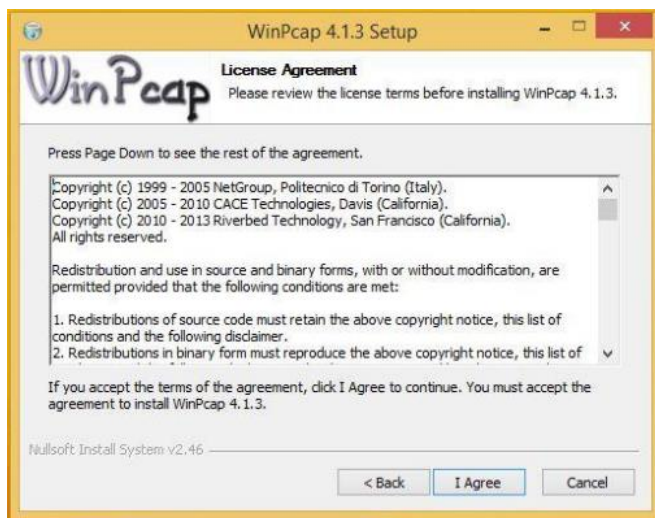


Рис. 4. Окно установки WinPcap

Для завершения процесса инсталляции программы нажмите Finish (Рис.5):



Рис. 5. Окно завершения установки

Первый запуск и начало работы с программой

При запуске программы она принимает следующий вид (Рис. 6):

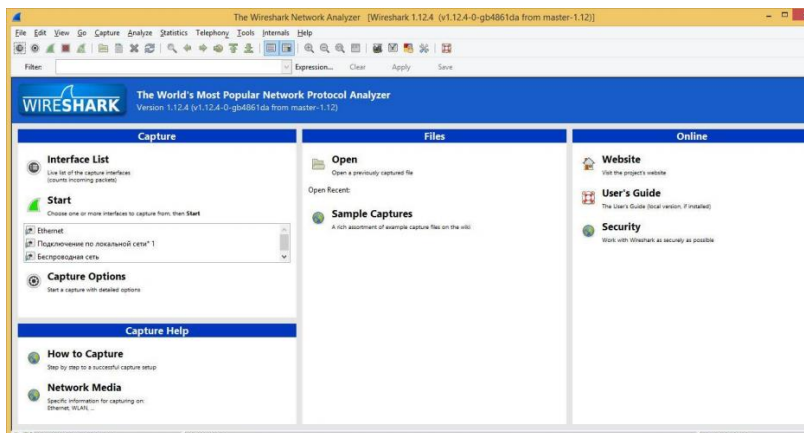


Рис. 6. Начальное окно программы

Перехват трафика является одной из ключевых возможностей [Wireshark](https://www.wireshark.org/). 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.

Для начала сборки перехваченных программой пакетов сообщений по сети, Вам необходимо выбрать пункт главного меню Capture>Interfaces или кнопку Interface List–после этого на экране появится следующее диалоговое окно (Рис. 7):



Рис. 7. Окно Interface List

С помощью кнопки Options возможна установка желаемых параметров работы программы. Для того, чтобы начать процедура захвата, Вам необходимо нажать кнопку Start, после чего интерфейс программы примет следующий вид (Рис. 8):

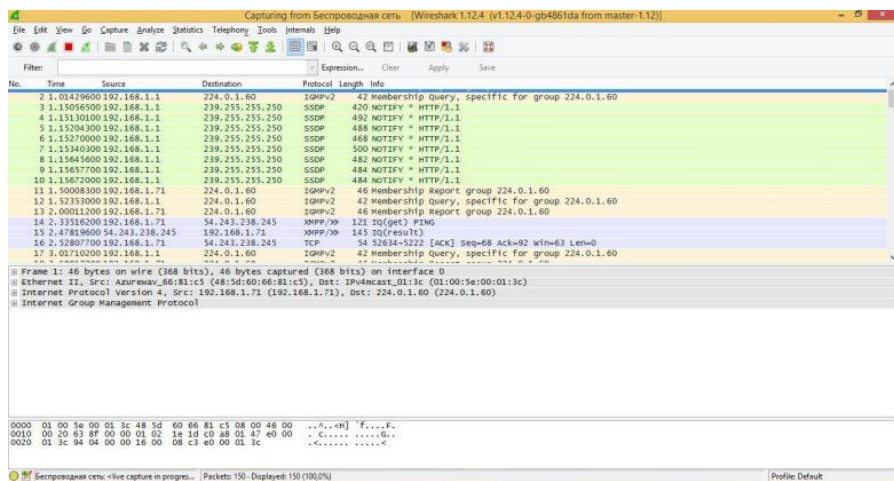


Рис. 8. Процедура захвата

Как видно из изображения, окно Wireshark включает в себя 3 области просмотра с различными уровнями детализации. Верхнее окно содержит список собранных пакетов с кратким описанием, в среднем окне показывается дерево протоколов, инкапсулированных в кадр. Ветви дерева могут быть раскрыты для повышения уровня детализации выбранного протокола. Последнее окно содержит дамп пакета в шестнадцатеричном или текстовом представлении. Программа Wireshark представляет пользователю ряд уникальных возможностей, не поддерживаемых другими анализаторами протоколов.

Программа обеспечивает возможность сбора всех пакетов заданного соединения ТСП и представления данных в удобном для просмотра формате.

Возможности программы

Рассмотрим возможности программы более подробно. В верхней панели по умолчанию выводится 6 колонок – номера пакета в списке собранных, временная метка, адреса и номера портов отправителя и получателя, тип протокола и краткое описание пакета.

Выбрав необходимый пакет из списка, содержащейся в верхней панели, мы можем просмотреть содержимое средней панели. В ней представлено дерево протоколов для пакета. Дерево отображает каждое поле и его значение для заголовков всех протоколов стека.

С помощью программы [Wireshark](#) возможно контролировать пакеты, проходящие по протоколу HTTP. Приведем пример вычисления имени пользователя и пароля при входе в почту. Для этого необходимо предварительно произвести процедуру входа в почтовый ящик и запуск программы в режиме захвата пакетов сообщений. После этого для удобства желательно произвести сортировку по протоколам при помощи нажатия на колонку Protocol. Далее, выбрав любой пакет, у которого установлено значение Protocol - HTTP, кликнуть правую кнопку мыши, в контекстном меню нажать Conversation filter>IP, тем самым выбрав фильтрацию списка пакетов только данного перечня адресов отправителя и получателя (Рис. 9):

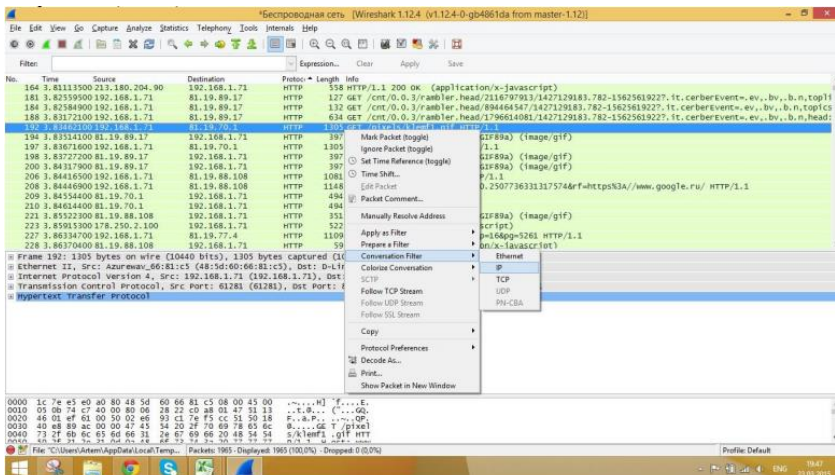


Рис. 9. Выбор фильтрации

Далее возможно отфильтровать пакеты сообщений, выбрав только пакеты протокола HTTP – для этого выберем любой пакет протокола HTTP и кликнем правую кнопку мыши и в контекстном меню выберем Conversation filter>Ethernet. Теперь у нас в верхнем окне остались только пакеты протокола HTTP. Для оптимизации поиска нужного пакета, воспользуемся пунктом главного меню Edit>Find Packet (Рис. 10):

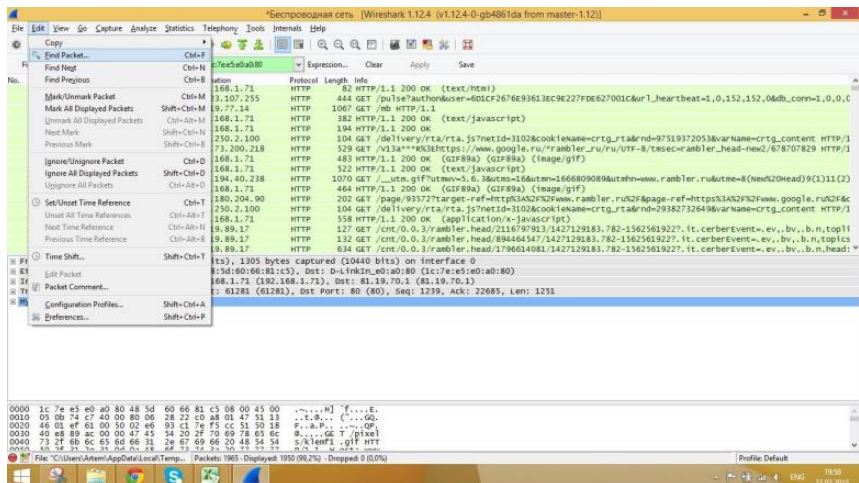


Рис. 10. Запуск поиска

В появившемся диалоговом окне выбираем Find by string и указываем название строки с адресом страницы почтового ящика (Рис. 11):

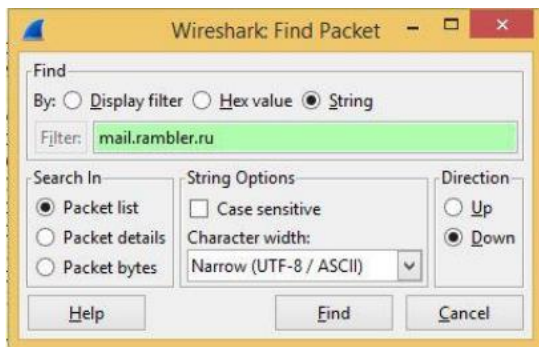


Рис. 11. Настройка поиска

Последний шаг - с помощью системы поиска проверяем пакеты, соответствующие адресации и в дереве протоколов находим в ветви Line-based text data значения Login, Domain, Password.

Wireshark предоставляет возможность пользователю сохранять файлы данных (изображения, CSS и др.) на жесткий диск из просмотренных ранее страниц в Интернете.

Для этого необходимо в главном меню программы выбрать File>Export>Objects>HTTP (Рис. 12):

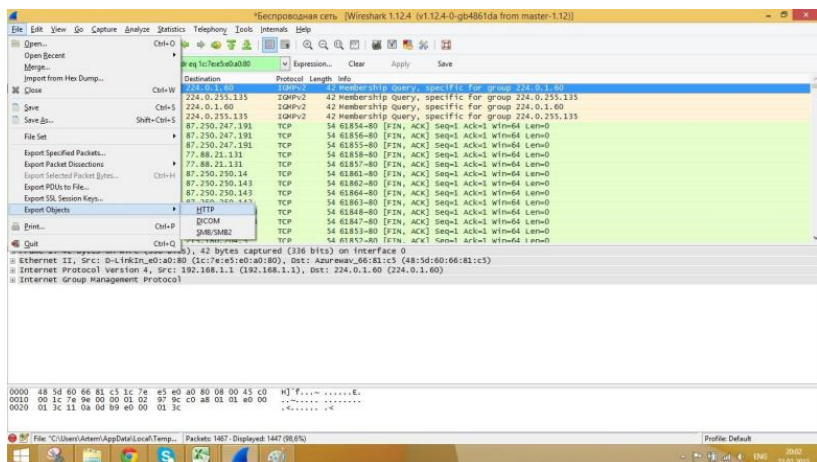


Рис. 12. Настройка сохранения изображения

В появившемся списке HTTP object list выбираем необходимый файл и нажимаем Save As (Рис. 13):

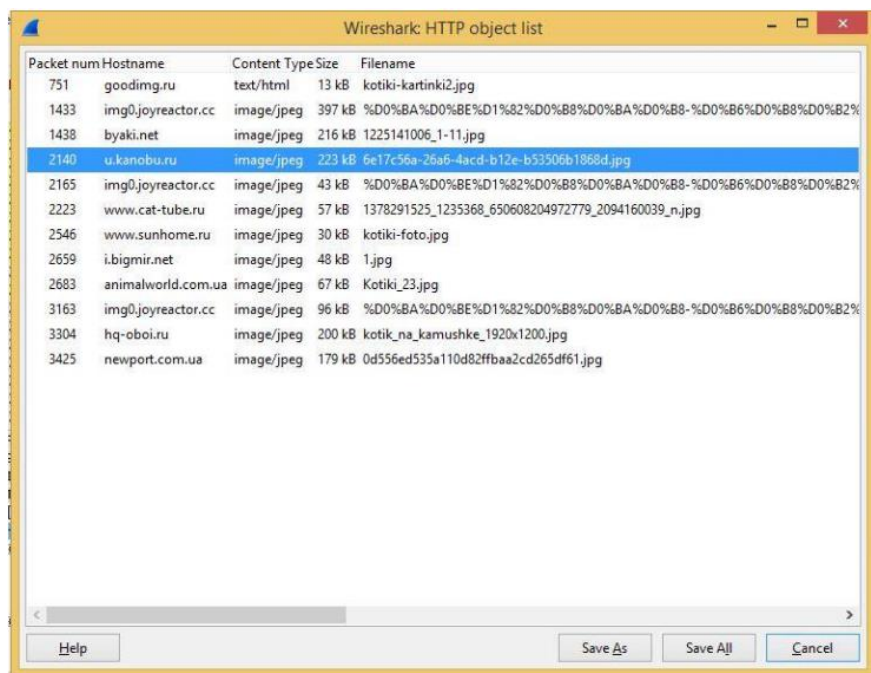


Рис. 13. Сохранение изображения

Далее программа предложит нам выбрать путь для сохранения файла на диск (Рис. 14):

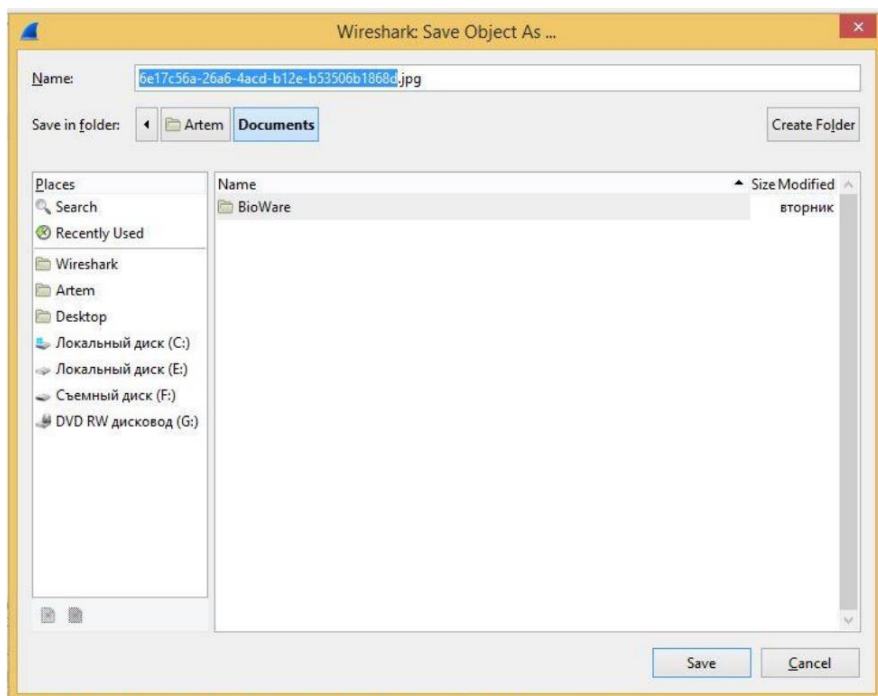


Рис. 14. Выбор места сохранения изображения

Дополнительные параметры

В анализаторе протоколов [Wireshark](#) возможна маркировка при помощи установки курсора на нужный пакет и выбора в контекстном меню после нажатия правой кнопки мыши Mark Packet (Рис. 15). Далее возможен быстрый поиск маркированного Вами пакета при помощи главного меню Edit>Find Next Mark:

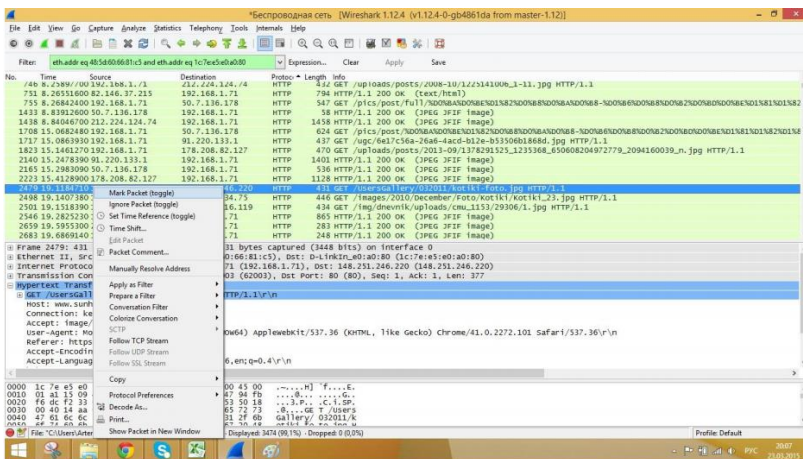


Рис. 15. Маркировка пакетов

Помимо сохранения передаваемых файлов, в программе также предусмотрена возможность экспорта суммарной информации о пакетах и дереве протоколов в файл формата .txt при помощи главного меню File>Export Packet Dissections >as “file Text” file (Рис. 16):

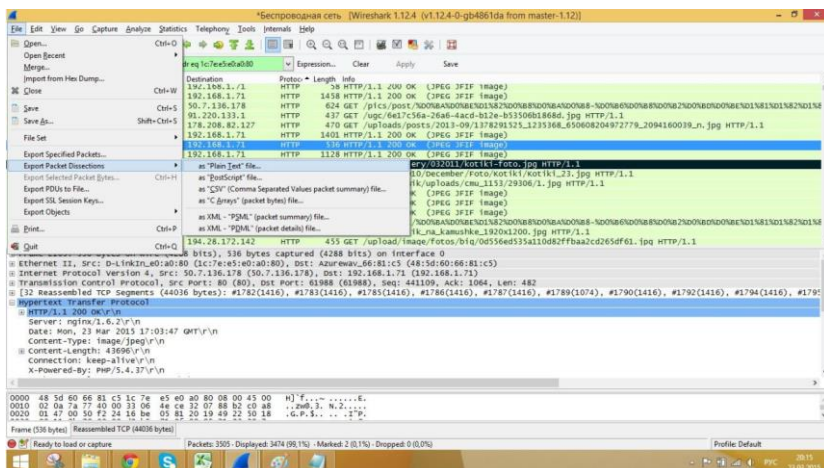


Рис. 16. Сохранение информационного файла

Полученный текстовый файл выглядит следующим образом (Рис. 17):

15 — БЛОКНОТ

Файл	Правка	Формат	Вид	Справка					
No.	Time	Source	Destination	Protocol	Length	Info			
1	0.000000000	192.168.1.71	64.233.165.188	TCP	55	60187→5228 [ACK] Seq=1 Ack=1 Win=63 Len=1			
Frame 1: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0									
Ethernet II, Src: Azurewav_66:81:c5 (48:5d:60:66:81:c5), Dst: D-LinkIn_e0:a0:80 (1c:7e:e5:e0:a0:80)									
Internet Protocol Version 4, Src: 192.168.1.71 (192.168.1.71), Dst: 64.233.165.188 (64.233.165.188)									
Transmission Control Protocol, Src Port: 60187 (60187), Dst Port: 5228 (5228), Seq: 1, Ack: 1, Len: 1									
Data (1 byte)									
0000 00									
No.	Time	Source	Destination	Protocol	Length	Info			
2	0.030024000	64.233.165.188	192.168.1.71	TCP	66	5228→60187 [ACK] Seq=1 Ack=2 Win=361 Len=0 SLE=1 SRE=2			
Frame 2: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0									
Ethernet II, Src: D-LinkIn_e0:a0:80 (1c:7e:e5:e0:a0:80), Dst: Azurewav_66:81:c5 (48:5d:60:66:81:c5)									
Internet Protocol Version 4, Src: 64.233.165.188 (64.233.165.188), Dst: 192.168.1.71 (192.168.1.71)									
Transmission Control Protocol, Src Port: 5228 (5228), Dst Port: 60187 (60187), Seq: 1, Ack: 2, Len: 0									

Рис. 17. Сохраненный файл

Кроме прочего программа обладает большим набором вывода статистических данных о захваченных пакетах сообщений. Так, можно вывести общую таблицу иерархии протоколов при помощи пункта главного меню Statistics>Protocol Hierarchy (Рис. 18):

Wireshark: Protocol Hierarchy Statistics									
Display filter: eth.addr eq 48:5d:60:66:81:c5 and eth.addr eq 1c:7e:e5:e0:a0:80									
Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s End	Packets End	Bytes End	Mbit/s	
Ethernet	100.00 %	3474	100.00 %	2565464	0,688	0	0	0,000	
Internet Protocol Version 4	99.94 %	3472	100.00 %	2565380	0,688	0	0	0,000	
Transmission Control Protocol	97.75 %	3396	99.62 %	2555766	0,685	2300	1738093	0,466	
Data	0.03 %	1	0.00 %	55	0,000	1	55	0,000	
Secure Sockets Layer	30.71 %	1067	31.23 %	801300	0,215	1017	735563	0,197	
Secure Sockets Layer	1.44 %	50	2.56 %	65737	0,018	50	65737	0,018	
XMPP Protocol	0.06 %	2	0.01 %	266	0,000	2	266	0,000	
Hypertext Transfer Protocol	0.69 %	24	0.60 %	15314	0,004	12	5697	0,002	
Line-based text data	0.03 %	1	0.03 %	794	0,000	1	794	0,000	
JPEG File Interchange Format	0.32 %	11	0.34 %	8823	0,002	11	8823	0,002	
Malformed Packet	0.06 %	2	0.03 %	738	0,000	2	738	0,000	
Internet Group Management Protocol	0.98 %	34	0.06 %	1440	0,000	34	1440	0,000	
Internet Control Message Protocol	0.06 %	2	0.05 %	1180	0,000	2	1180	0,000	
User Datagram Protocol	1.15 %	40	0.27 %	6994	0,002	0	0	0,000	
Domain Name Service	0.86 %	30	0.11 %	2922	0,001	30	2922	0,001	
Teredo IPv6 over UDP tunneling	0.06 %	2	0.01 %	254	0,000	0	0	0,000	

Рис. 18. Таблица иерархии протоколов

Для наглядного представления результатов выполнения захвата пакетов и сборки кадров в программе имеется возможность отображения данной информации в виде графика передачи пакетов в единицу времени. Для отображения данного графика необходимо воспользоваться пунктом главного меню Statistics>IO Graphs (Рис. 19):

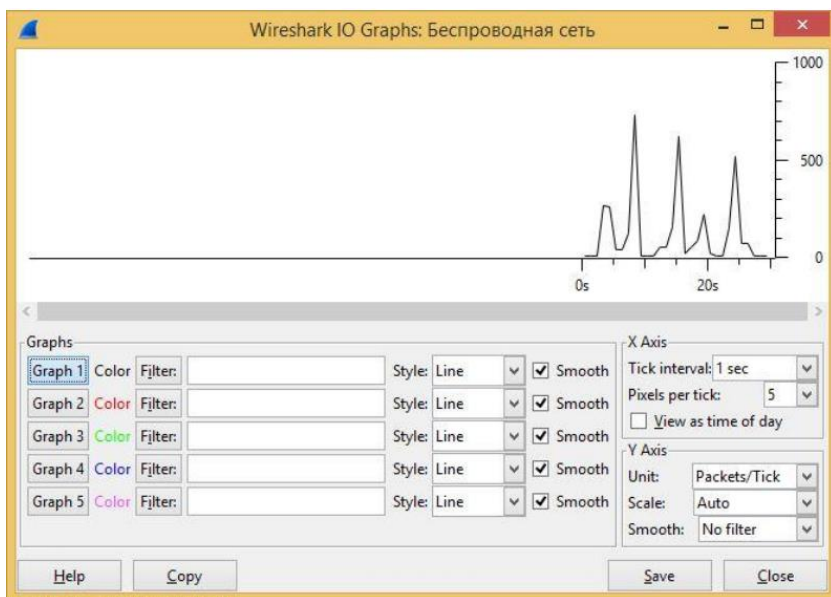


Рис. 18. График передачи пакетов в единицу времени

Заключение

Проблема сниффинга была актуальной раньше – в сетях, основанных на концентраторах (хабах) – она остается актуальной и сейчас для сетей на коммутаторах (свитчах), благодаря такой технологии как ARP spoofing. Более того, сегодня семимильными шагами развивается технологии беспроводных сетей, где сниффинг возможен даже в пассивном режиме.

Единственным решением, препятствующим сниффингу, является шифрование. Нельзя допускать использования фирменных небезопасных прикладных протоколов или унаследованных протоколов, передающих данные явным образом. Замена небезопасных протоколов (таких как telnet) на их надежные зашифрованные аналоги (такие как SSH) представляется серьезным барьером от перехвата. Замена всех небезопасных протоколов в большинстве случаев маловероятна.

Вместо прекращения использования протоколов, передающих данные явным образом, остается только одна возможность -

шифрование всего сетевого трафика на 3 уровне, используя IPSec. Осуществляя шифрование на 3 уровне, возможно продолжать использовать небезопасные протоколы, поскольку все данные будут инкапсулированы IPSec и зашифрованы при передаче по сети. Таким образом, унаследованные приложения, которые используют старые протоколы, не пострадают. IPSec полностью прозрачен для приложений и пользователей. Это открытый стандарт, поддерживаемый многими вендорами, включая Microsoft и Cisco. Кроме того, многие реализации Unix поддерживают IPSec. Легкая настройка IPSec в Win2k/XP дополнительно увеличивает его доступность.

Осуществление технологии шифрования на 3 уровне, таких как IPSec решает проблему сниффинга полностью. Масштабируемость, распространенность, доступность IPSec выделяет его как прагматическое решение проблемы перехвата сетевого трафика

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1. Ознакомиться с предложенным теоретическим материалом для получения информации о методах сбора и анализа передаваемых данных.
2. Применить на практике полученные знания в виде собранной статистики по передаваемым данным.
3. Подготовить ответы на контрольные вопросы.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Дайте определение термину «снифер».
2. Перечислите способы перехвата трафика.
3. Изложите что позволяет анализ прошедшего через снифер трафика.
4. Перечислите на какие категории разделяют сниферы.
5. Дайте определение термину Wireshark. Перечислите для чего он используется, возможности, что Wireshark делать не умеет.
6. Перечислите возможности Wireshark по перехвату.
7. Перечислите основные этапы установки Wireshark.
8. Определите мешает ли шифрование снифингу.
9. Изложите какой протокол необходимо использовать при осуществлении шифрования всего сетевого трафика на 3 уровне.
10. Опишите области просмотра окна Wireshark.
11. Опишите библиотеку Pcap.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 2 занятия (4 академических часа: 3 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания (вариант), этапы выполнения работы (со скриншотами), результаты выполнения работы. выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Смелянский Р.Л. Компьютерные сети. 2 т. Т.1. Системы передачи данных [Текст] / Р.Л. Смелянский. –М.: Изд. Центр «Академия», 2011.- 304 с.
2. Смелянский Р.Л. Компьютерные сети. В 2 т. Т.2. Сети ЭВМ [Текст]: учебник для вузов / Р.Л. Смелянский. –М.: Изд. Центр «Академия», 2011.- 240 с.
3. Власов Ю.В. Администрирование сетей на платформе MS Windows Server [Электронный ресурс] / Ю.В. Власов, Т.И. Рицкова. —М.: Интернет-Университет Информационных технологий (ИНТУИТ), 2016. — 622 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/52219.html>

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

4. Технологии коммутации и маршрутизации в локальных компьютерных сетях. [Текст]: учеб. пособие для вузов / Е.В. Смирнова, А.В. Пролетарский [и др.]; под. ред. А.В. Пролетарского. -М.: Изд-во МГТУ им. Н.Э. Баумана, 2013. - 389 с.: ил.
5. Таненбаум Э. Компьютерные сети [Текст] / Э. Таненбаум. – 4-е изд. – СПб.: Питер, 2010. — 992 с.
6. Ачилов Р.Н. Построение защищенных корпоративных сетей [Электронный ресурс]: учеб. пособие / Р.Н. Ачилов. — Москва: ДМК Пресс, 2013. — 250 с. — 2227-8397. — Режим доступа: <http://e.lanbook.com/book/66472>

Электронные ресурсы:

7. Электронно-библиотечная система «Лань»
8. Электронно-библиотечная система «Университетская библиотека ONLINE»
9. Электронно-библиотечная система «IPRbooks»
10. Электронно-библиотечная система «Юрайт»