Лабораторная работа № 5

Исследование локальных и глобальных сетей, протоколов сетевого уровня (IP, ICMP) с использованием анализаторов (снифферов).

Цель работы

Знакомство с анализаторами сетевого трафика (снифферами) для исследования локальных и глобальных сетей и сетевого трафика. Анализ и исследование IP трафика на основе наблюдений за обменом датаграммами узлов, анализ формата IP пакетов.

Теоретические сведения

Протокольный стек TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) — это промышленный стандарт стека протоколов, разработанный для глобальных сетей.

Стандарты TCP/IP опубликованы в серии документов, названных Request for Comment (RFC). Документы RFC описывают внутреннюю работу сети Internet. Некоторые RFC описывают сетевые сервисы или протоколы и их реализацию, в то время как другие обобщают условия применения. На этом стеке работает сеть Internet.

Свойства стека TCP/IP:

- Это наиболее завершенный стандартный и популярный стек сетевых протоколов.
- Все сети передают основную часть трафика с помощью протоколов TCP/IP.
- Это метод получения доступа к сети Internet.
- Стек служит основой для создания корпоративных сетей, использующих транспортные услуги Internet.
- Все операционные системы поддерживают стек TCP/IP.
- Это технология для соединения разнородных систем как на уровне транспортных подсистем, так и на уровне прикладных сервисов.
- Это масштабируемая межплатформенная среда для приложений клиент-сервер.

Стек TCP/IP имеет многоуровневую структуру и делится на 4 уровня. В таблице приведено примерное соответствие уровней TCP/IP уровням модели OSI.

Уровень модели OSI		Примеры протоколов	Уровень TCP/IP
7	Прикладной уровень	WWW, SNMP, FTP, telnet, SMTP, TFTP	1
6	Представительский уровень		
5	Сеансовый уровень	TCP, UDP	2
4	Транспортный уровень		
3	Сетевой уровень	IP, ICMP, RIP, OSPF, ARP	3
2	Канальный уровень	Ethernet, PPP, FDDI, X.25	4
1	Физический уровень		

Самый нижний (уровень IV) в протоколах TCP/IP не регламентируется, но поддерживает все стандарты физического и канального уровня: для локальных сетей это Ethernet, Token Ring, FDDI, Fast Ethernet, 100VG-AnyLAN, для глобальных сетей — протоколы соединений "точка-точка" SLIP и PPP, протоколы территориальных сетей с

коммутацией пакетов X.25, frame relay. Разработана также специальная спецификация, определяющая использование технологии ATM в качестве транспорта канального уровня. При появлении новой технологии локальных или глобальных сетей она включается в стек TCP/IP за счет разработки соответствующего RFC, определяющего метод инкапсуляции пакетов IP в ее кадры.

Следующий уровень (уровень III) — это уровень межсетевого взаимодействия, который занимается передачей пакетов с использованием различных транспортных технологий локальных сетей, территориальных сетей, линий специальной связи и т. п.

В качестве основного протокола сетевого уровня (в терминах модели OSI) в стеке используется протокол IP. Протокол IP работает в сетях со сложной топологией и рационально использует пропускную способность низкоскоростных линий связи. Протокол IP является дейтаграммным протоколом, то есть он не гарантирует доставку пакетов до узла назначения.

К уровню межсетевого взаимодействия относятся протоколы, связанные с составлением и модификацией таблиц маршрутизации, такие как протоколы сбора маршрутной информации RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First), а также протокол межсетевых управляющих сообщений ICMP (Internet Control Message Protocol).

Следующий уровень (уровень II) называется основным. На этом уровне функционируют протокол управления передачей TCP (Transmission Control Protocol) и протокол дейтаграмм пользователя UDP (User Datagram Protocol). Протокол TCP обеспечивает надежную передачу сообщений между удаленными прикладными процессами за счет образования виртуальных соединений. Протокол UDP обеспечивает передачу прикладных пакетов дейтаграммным способом и выполняет только функции связующего звена между сетевым протоколом и прикладными процессами.

Верхний уровень (уровень I) называется прикладным. TCP/IP накопил большое количество протоколов и сервисов прикладного уровня. К ним относятся такие протоколы, как протокол копирования файлов FTP, протокол эмуляции терминала telnet, почтовый протокол SMTP, используемый в электронной почте сети Internet, гипертекстовый сервис доступа к информации WWW и многие другие.

IP (Internet Protocol)

Протокол IP создан для использования в объединенных системах компьютерных сетей с коммутацией пакетов. Протокол обеспечивает передачу блоков данных, называемых датаграммами, от отправителя к получателям, где отправители и получатели являются узлами, идентифицируемыми адресами фиксированной длины. Протокол Internet обеспечивает при необходимости фрагментацию и сборку датаграмм для передачи данных через сети с малым размером пакетов.

Протокол ограничен задачами обеспечения функций, необходимых для передачи битового пакета (датаграммы). Он не имеет механизмов для увеличения достоверности данных, управления протоколом, синхронизации или других услуг.

Две его функции:

- адресация,
- фрагментация.

Адреса помещаются в заголовок IP пакета для передачи датаграмм получателям. Выбор пути передачи называется маршрутизацией.

Также IP протокол использует поля в заголовке IP пакета для фрагментации и восстановления датаграмм для передачи через сети с малым размером пакетов. Протокол реализует общие правила для интерпретации полей адресов, фрагментации и сборки датаграмм.

Протокол обрабатывает каждую датаграмму как независимую единицу, не имеющую связи ни с другими датаграммами. Протокол не имеет дело ни с соединениями, ни с логическими цепочками (виртуальными или какими-либо другими).

Протокол Internet использует четыре механизма для формирования услуг:

- тип сервиса,
- время жизни,
- опции,
- контрольная сумма заголовка.

Тип сервиса используется для обозначения требуемой услуги. Это набор параметров, который характеризует набор услуг, предоставляемых сетями. Они используются шлюзами для выбора рабочих параметров передачи в конкретной сети, для выбора сети, используемой при следующем переходе датаграммы, для выбора следующего шлюза при маршрутизации датаграммы.

Механизм времени жизни служит для указания верхнего предела времени жизни датаграммы. Этот параметр устанавливается отправителем и уменьшается в каждой точке на проходимом маршруте. Если параметр времени жизни станет нулевым до того, как датаграмма достигнет получателя, то датаграмма будет уничтожена.

Механизм опций предоставляет функции управления, которые являются необходимыми или полезными при определенных ситуациях, однако он не нужен при обычных коммуникациях. Механизм опций предоставляет такие возможности, как временные штампы, безопасность, специальная маршрутизация.

Контрольная сумма заголовка обеспечивает проверку того, что информация, используемая для обработки датаграмм, передана правильно. Если контрольная сумма неверна, то датаграмма будет разрушена, как только ошибка будет обнаружена.

Протокол Internet не обеспечивает надежности коммуникации, так как не имеет механизм подтверждений между отправителем и получателем, между узлами, не имеет контроля ошибок для поля данных, не поддерживает повторной передачи и управления потоком.

Схема движения данных

Данные по сети передаются в три этапа:

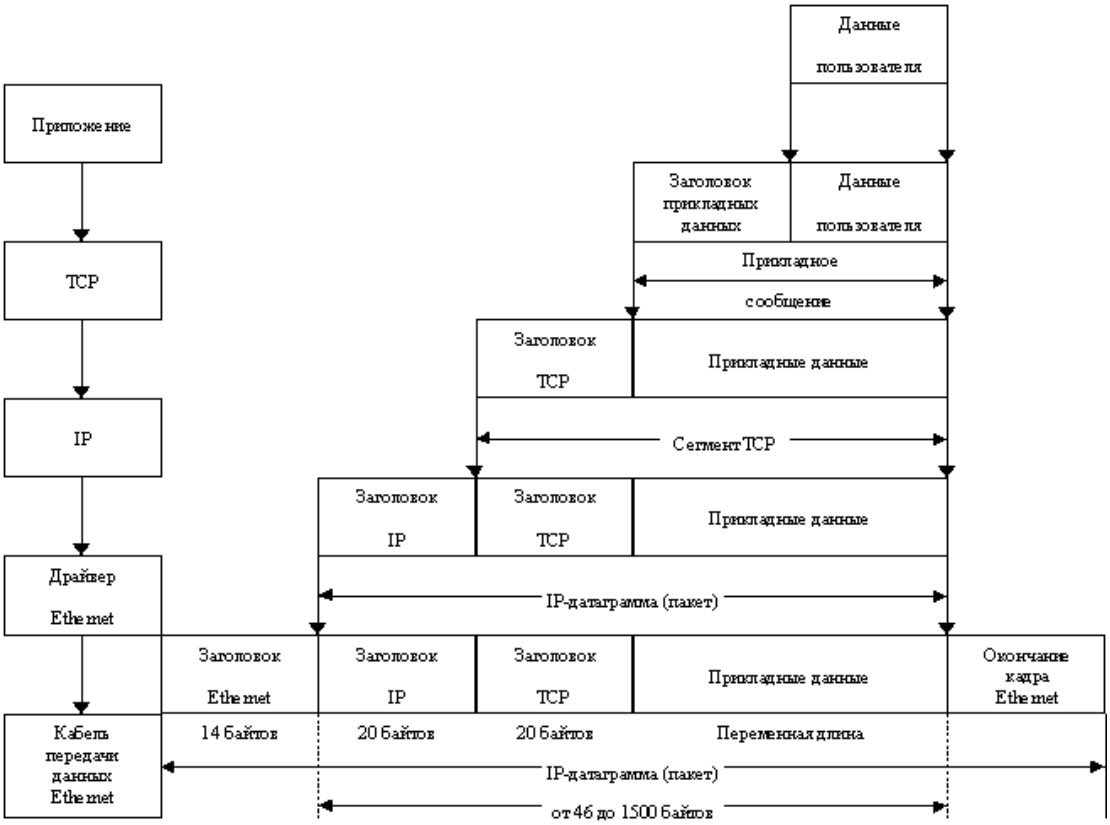
1. Информация должна пройти между приложениями и сетью. Это путь сквозь стек протоколов вниз к транспортному уровню.
2. Определение сетью адреса получателя данных.
3. Маршрутизация данных и прохождение данных сквозь стек протоколов вверх к сетевому приложению.

Схема движения данных пользователя представлена на рисунке.

Схему действий при передаче датаграммы от одной прикладной программы к другой можно проиллюстрировать следующим образом.

Предположим, что перенос будет включать прохождение одного промежуточного шлюза.

1. Отправляющая прикладная программа готовит свои данные и вызывает свой локальный модуль IP протокола для отправки данных в качестве датаграммы, а в качестве аргументов этого вызова передает адрес получателя и другие параметры.
2. Модуль готовит заголовок датаграммы и стыкует с ним данные. Он определяет локальный сетевой адрес, соответствующий данному адресу Internet. В данном случае это адрес шлюза.
3. Модуль передает данную датаграмму и адрес в локальной сети в распоряжение интерфейса локальной сети.
4. Интерфейс локальной сети создает соответствующий этой сети заголовок и соединяет с ним датаграмму. Затем он передает по локальной сети полученный таким образом результат.



5. Датаграмма достигает узла, играющего роль шлюза. Интерфейс локальной сети отделяет этот заголовок и передает датаграмму модулю IP протокола.

6. Модуль определяет из IP адреса, что датаграмма должна быть направлена на узел во второй сети и определяет адрес узла-получателя.

7. Он обращается к интерфейсу локальной сети с тем, чтобы она переслала датаграмму по назначению.

8. Интерфейс создает заголовок локальной сети и соединяет с ним датаграмму, а затем результат направляет на узел-получатель.

9. На узле-получателе интерфейс локальной сети удаляет заголовок локальной сети и передает оставшееся на IP модуль.

10. Модуль Internet определяет, что датаграмма предназначена для прикладной программы на этом узле и передает данные прикладной программе в ответ на системный вызов. В качестве результата вызова передаются адрес получателя и другие параметры.

IP адресация

В IP протоколе сделано разграничение между именами, адресами и маршрутами. Имя показывает искомый объект. Адрес показывает его местонахождение.

Перевод имен в адреса является задачей протоколов прикладного уровня. IP протокол осуществляет отображение IP адресов на MAC адреса локальной сети.

Адреса протокола 4 версии (IP-4) имеют фиксированную длину четыре октета (32 бита). Обычно, он записывается в виде X.X.X.X, где каждая позиция, отделенная точкой — десятичное значение октета.

IP-адрес состоит из двух частей: номера сети и номера узла.

Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации подразделения Network Information Center, если сеть должна работать как составная часть Internet. Провайдеры услуг Internet получают диапазоны адресов у подразделений NIC, а затем распределяют их между своими абонентами.

Номер узла в протоколе IP назначается независимо от локального адреса узла. Деление IP-адреса на поле номера сети и номера узла — гибкое, и граница между этими

полями может устанавливаться весьма произвольно. Один узел может входить в несколько IP-сетей. В этом случае узел должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не компьютер или маршрутизатор, а сетевой интерфейс.

IP-адрес имеет длину 4 байта и записывается в десятичной или двоичной форме, например:

128.10.2.30 —десятичная форма представления адреса,

10000000 00001010 00000010 00011110 — двоичная форма представления адреса.

Диапазоны номеров сетей, соответствующих классу приведены в таблице.

Класс	Наименьший адрес	Наибольший адрес
A	01.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.1.0	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

В протоколе IP существует несколько соглашений об интерпретации IP-адресов:

- Если IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет.
- Если в поле номера сети стоят двоичные нули, то по умолчанию считается, что этот узел принадлежит той же самой сети, что и узел, который отправил пакет.
- Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным сообщением (limited broadcast).
- Если в поле адреса назначения стоят сплошные 1, то пакет рассылается всем узлам сети с заданным номером. Такая рассылка называется широковещательным сообщением (broadcast).
- Адрес 127.0.0.1 зарезервирован для организации обратной связи при тестировании работы программного обеспечения узла без реальной отправки пакета по сети. Этот адрес имеет название loopback.

Форма группового IP-адреса multicast означает, что пакет должен быть доставлен нескольким узлам, которые образуют группу с номером, указанным в поле адреса.

Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп.

Фрагментация

Фрагментация датаграммы необходима, когда эта датаграмма возникает в локальной сети, позволяющей работать с пакетами большого размера, и затем должна пройти к получателю через другую локальную сеть, которая ограничивает пакеты меньшим размером.

Датаграмма может быть помечена как нефрагментируемая. Любая датаграмма, помеченная таким образом, не может быть фрагментирована. Если датаграмма, помеченная как не фрагментируемая, не может достигнуть получателя без фрагментации, то вместо этого она будет разрушена.

Чтобы разделить датаграмму, модуль протокола IP создает две новые датаграммы и копирует содержимое полей заголовка из большой датаграммы в оба новых заголовка. Данные из старой датаграммы делятся на две части. Первая часть данных помещается в первую новую датаграмму, в поле общей длины помещается длина первой датаграммы. Флаг "more fragments" устанавливается в единицу. Вторая часть данных помещается во вторую датаграмму, в поле общей длины заносится длина второй датаграммы. В поле

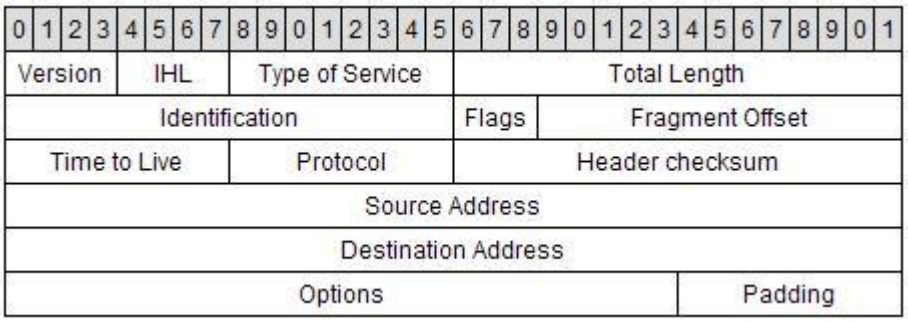
смещения фрагмента во второй датаграмме устанавливается значение относительно такого же поля в исходной большой датаграмме.

Эта процедура может быть обобщена на случай многократного расщепления исходной датаграммы.

Чтобы собрать фрагменты датаграммы, модуль протокола объединяет датаграммы, имеющие одинаковые значения в полях идентификатора, отправителя, получателя и протокола. Собственно объединение заключается в помещении данных из каждого фрагмента в позицию, указанную в заголовке пакета в поле "fragment offset". Первый фрагмент будет иметь в поле "fragment offset" нулевое значение, а последний фрагмент будет иметь флаг "more fragments", вновь установленный в нуль.

Формат IP пакета

На рисунке приведена схема полей заголовка IP пакета.



Каждая позиция соответствует одному биту.

1. Version (версия) 4 бита.

Поле версии показывает формат заголовка Internet. В работе используется IP версии 4.

2. IHL (длина Internet заголовка).

Длина Internet заголовка измеряется в словах по 32 бита каждый и указывает на начало поля данных. Корректный заголовок может иметь минимальный размер 5 слов. Чаще всего длина IP-заголовка равна 20 байтам, поэтому данное поле обычно содержит число 5 (0101).

3. Type of Service (тип обслуживания) 8 бит.

Тип обслуживания определяет с помощью неких абстрактных параметров тип требуемого обслуживания. Эти параметры должны использоваться для управления выбором реальных рабочих характеристик при передаче датаграммы через конкретную сеть. Некоторые сети осуществляют обслуживание с приоритетом, которое неким образом дает преимущество для продвижения данной датаграммы по сравнению со всеми остальными. Формат данного поля показан на рисунке.

Первое трехразрядное субполе приоритет(precedence) редко применяется на практике. Последнее безымянное одноразрядное субполе всегда содержит 0. Между ними находятся четыре одноразрядных субполя, которые и называют собственно битами TOS. Каждому из четырех битов TOS сопоставлен определенный критерий доставки дейтаграмм: минимальная задержка, максимум пропускной способности, максимум надежности и минимум стоимости. Только один бит TOS может быть установлен в 1. По умолчанию все четыре бита равны 0, что означает отсутствие особых требований, то есть обычный сервис. Реально выбор осуществляется между тремя альтернативами: малой задержкой, высокой достоверностью и высокой пропускной способностью.

4. Total Length (общая длина) 16 бит.

Общая длина — это длина датаграммы, измеренная в октетах, включая Internet заголовок и поле данных. Это поле может задавать длину датаграммы вплоть до 65535 октетов. В большинстве случаев столь большие датаграммы не используются. Все узлы должны быть готовы принимать датаграммы вплоть до 576 октетов длиной (приходят ли они целиком или по фрагментам). Узлам рекомендуется отправлять датаграммы размером более чем 576 октетов, только если они уверены, что принимающий узел готов обслуживать датаграммы повышенного размера. Если локальная сеть построена по технологии Ethernet, уровень соединения инкапсулирует IP-дейтаграммы в кадры Ethernet перед передачей их в Интернет. Спецификация Ethernet ограничивает длину пакета до 1500 байтов.

- 5. **Identification** (идентификатор) 16 бит.

Идентификатор устанавливается отправителем для сборки фрагментов какой-либо датаграммы.

6. Flags (контрольные биты) 3 бита.

Ниже приведена таблица значений контрольных битов.

Смещение	Значение
0	Зарезервирован, всегда должен быть ноль
1	0 — возможно фрагментирование 1 — запрет фрагментации
2	0 — последний фрагмент или нет фрагментации 1 — будут еще фрагменты

7. Fragment Offset (смещение фрагмента) 13 бит.

Это поле показывает, где в датаграмме находится этот фрагмент. Смещение фрагмента изменяется порциями по 8 октет (64 бита). Первый фрагмент имеет смещение ноль.

8. Time to Live (Время жизни) 8 бит.

Это поле показывает максимальное время, в течении которого датаграмме позволено находиться в системе Internet. Если это поле имеет значение нуль, то датаграмма должна быть разрушена. Значение этого поля изменяется при обработке заголовка Internet. Время измеряется в секундах. Однако, поскольку каждый модуль, обрабатывающий датаграмму, должен уменьшать значение поля TTL по крайней мере на единицу, даже если он обрабатывает эту датаграмму менее, чем за секунду, то поле TTL следует понимать как максимальный интервал времени, в течении которого датаграмма может существовать.

9. Protocol (Протокол) 8 бит.

Это поле показывает, какой протокол следующего уровня использует данные из Internet датаграммы. Например, при использовании TCP данное поле будет принимать значение 6.

10. Header Checksum (Контрольная сумма заголовка) 16 бит.

Поскольку некоторые поля заголовка меняют свое значение (например, время жизни), это значение проверяется и повторно рассчитывается при каждой обработке Internet заголовка.

11. Source Address (адрес отправителя) 32 бита.

12. Destination Address (адрес получателя) 32 бита.

13. Options (опции) поле переменной длины.

Опции могут появиться в датаграммах, а могут и не появляться. Они должны поддерживаться всеми Internet модулями. Не обязательно каждая конкретная датаграмма несет опции, но нести их все же может.

Анализаторы сетевого трафика

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.

Сниффер может анализировать только то, что проходит через его сетевую карту. Внутри одного сегмента сети Ethernet все пакеты рассылаются всем машинам, из-за этого возможно перехватывать чужую информацию. Использование коммутаторов (switch, switch-hub) и их грамотная конфигурация уже является защитой от прослушивания. Между сегментами информация передаётся через коммутаторы. Коммутация пакетов — форма передачи, при которой данные, разбитые на отдельные пакеты, могут пересылаться из исходного пункта в пункт назначения разными маршрутами. Так что если кто-то в другом сегменте посылает внутри него какие-либо пакеты, то в ваш сегмент коммутатор эти данные не отправит.

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 сниффер попадают лишь отдельные фреймы);

- подключением сниффера в разрыв канала;

- ответвлением (программным или аппаратным) трафика и направлением его копии на сниффер (Networktap);

- через анализ побочных электромагнитных излучений и восстановление таким образом прослушиваемого трафика;

- через атаку на канальном (2) (MAC-spoofing) или сетевом (3) уровне (IP-spoofing), приводящую к перенаправлению трафика жертвы или всего трафика сегмента на сниффер с последующим возвращением трафика в надлежащий адрес.

В 1990-х годах перехват траффика применялся хакерами для захвата пользовательских логинов и паролей, которые в ряде сетевых протоколов передаются в незашифрованном или слабозашифрованном виде. Широкое распространение хабов позволяло захватывать трафик без больших усилий в больших сегментах сети практически без риска быть обнаруженным.

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 (а порой 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)

5) Поскольку в «классическом» сниффере 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.

Анализатор WinDump

Общий вид инструкции запуска WinDump: WinDump: опции выражения
Опции WinDump перечислены в таблице

Опция	Описание
-a	Пытается преобразовать адреса в имена. Это создает дополнительную нагрузку на систему и может привести к потере пакетов
-c	Число. Останавливает WinDump после обработки заданного числа пакетов
-C	Размер_файла. Ограничивает размер выходных файлов заданным числом байт
-d	Выдает процедуру сопоставления пакетов с образцом в удобочитаемом виде и затем останавливается
-dd	Выдает процедуру сопоставления пакетов с образцом в виде фрагмента программы на языке Си
-ddd	Выдает процедуру сопоставления пакетов с образцом в виде десятичных чисел
-e	В каждой строке выдачи печатает заголовок канального уровня (в сетях Ethernet это MAC-адрес)
-E	Алгоритм «секрет». Использует встроенную в WinDump возможность расшифровывать на лету пакеты, зашифрованные по протоколу IPsecESP. Разумеется, чтобы использовать эту опцию, нужно располагать разделяемым секретным ключом. В число возможных значений параметра "алгоритм" входят des-cbc, 3des-cdc, blowfish-cbc, r3c-cbc, приведенный 128-cbc. Кроме того, оно может быть пустым. По умолчанию используется des-cbc. Значением параметра "секрет" должен служить секретный ключ ESP в текстовом виде
-F	Файл. Использует файл (а не сеть) для ввода данных. Это удобно для анализа событий "постфактум".
-i	Интерфейс. Читает из заданного интерфейса, когда на анализирующей машине имеется несколько сетевых интерфейсов. По умолчанию Win-Dump использует действующий интерфейс с наименьшим номером. В системах Linux можно использовать также параметр any для перехвата пакетов из всех сетевых интерфейсов
-n	Не преобразовывает адреса в имена
-N	Не печатает в именах хостов имя домена вышележащего уровня. Это полезно, если вам необходимо представить обезличенную версию вывода и вы не хотите раскрывать, чья это сеть
-p	Не переводит интерфейс в режим прослушивания. Используется только при исследовании трафика, направленного в анализирующий компьютер
-q	Печатает быстрый вывод. Печатается меньше протокольной информации, поэтому строки оказываются короче
-T	Тип. Заставляет интерпретировать пакеты, выбранные заданным в выражении фильтром, в соответствии с указанным типом
-t	Не печатает метку времени в каждой строке
-tt	Печатает неформатированную метку времени в каждой строке
-ttt	Печатает интервал времени между пакетами
-tttt	Печатает в каждой строке дату, а затем метку времени в подразумеваемом формате
-v	Использует чуть более подробный вывод, включающий время жизни, идентификатор, общую длину и поля опций каждого пакета
-vv	Предоставляет более детальный вывод. Пакеты NFS и SMB полностью декодируются
-vvv	Предоставляет еще более подробный вывод. Это может существенно за-медлить работу анализатора
-w	Имя_файла. Записывает пакеты в указанный файл вместо вывода их на экран. Таким образом результаты "вынюхивания" без участия человека можно сохранить и проанализировать их позже. Например, если в вашей сети происходят какие-то странные вещи, вы можете запустить WinDump на ночь, чтобы перехватить весь необычный трафик. Не забудьте написать хороший фильтр, иначе полученный наутро файл может оказаться слишком большим
-x	Выводит каждый пакет (без заголовка канального уровня) в шестнадцатеричном виде.
-X	Выводит содержимое пакетов и в шестнадцатеричном, и в текстовом видах

Выражения WinDump определяют выбор отображаемых сетевых пакетов. Именно здесь происходит реальная работа WinDump. Выдаются только те объекты, которые соответствуют выражению; если выражения не заданы, отображаться будут все пакеты. Выражение WinDump состоит из одной или нескольких директив, называемых

примитивами, которые, в свою очередь, состоят из идентификатора и следующего за ним квалификатора.

Существуют также более сложные выражения, которые можно строить с помощью булевых операций, таких как И, ИЛИ, НЕ, и операций сравнения (больше, меньше и т.п.). Ниже перечислены три различных вида квалификаторов, и доступные комбинации примитивов.

Квалификаторы WinDump

- тип: Определяет, к чему относится идентификатор, заданный как имя или номер. Возможными типами служат host, net и port. Например, hostfoo, net 128.3 или port 20

- направление: Определяет направление трафика от определенного идентификатора. Возможными направлениями служат src; dst; srcordst и srcanddst (src обозначает исходный адрес, dst - целевой)

- протокол: Позволяет определить протокол для фильтрации. Возможными протоколами являются ether, fddi, tr, ip, ipv6, arp, rarp, decnet, tcp и udp. Если протокол не задан, то допустимы все протоколы, совместимые с остальной частью выражения. При помощи фильтров с этим квалификатором можно определить, какая машина делает чрезмерное количество arp-запросов, или для отбрасывания на фильтре udp-запросов, которых немало во многих сетях, так как DNS использует udp

Допустимые комбинации примитивов

1) dsthost хост: Показывает только трафик, адресованный хосту, который может быть задан IP-адресом или именем;

2) srchost хост: Показывает только трафик, исходящий из хоста;

3) host хост: Показывает как исходящий, так и входящий трафик хоста;

4) etherdstEthernet-хост: Показывает трафик, предназначенный для указанного Ethernet-хоста, который может быть задан либо именем, либо MAC-адресом;

5) ethersrcEthernet-хост: Показывает трафик, исходящий из Ethernet-хоста

6) etherhostEthernet-хост: Показывает как исходящий, так и входящий трафик Ethernet-хоста;

7) gateway хост: Показывает любой трафик, использующий хост в качестве шлюза. Иными словами, трафик, переправляемый с хоста. Так происходит, когда IP-адрес отправителя или получателя не соответствует Ethernet-адресу хоста. Данную возможность целесообразно использовать, когда необходимо отследить весь трафик, проходящий через Интернет-шлюз или некоторый конкретный маршрутизатор;

8) dstnet сеть: Фильтрует трафик, предназначенный для конкретной сети, заданной в нотации 0.0.0.0. Аналогично etherdstEthernet-хост за исключением того, что это может быть значительно больше, чем один хост;

9) srcnet сеть: Фильтрует сеть отправителя;

10) net сеть: То же, что и две предыдущие инструкции, но трафик разрешен как в заданную сеть, так и из нее;

11) net сеть maskмаска_сети: Сопоставляется с трафиком в заданную сеть или из нее, с указанной маской сети. Применяется для задания точного размера сети с шагом меньше, чем класс C. В этой комбинации допускается использование примитивов src и dst для указания направления потоков данных;

12) net сеть/длина_маски: Сопоставляется с трафиком с сетевыми адресами из указанной сети и заданным числом бит в маске сети. Аналогична предыдущей комбинации;

13) dstport порт: Фильтрует трафик TCP и UDP с заданным целевым портом. Здесь можно также специфицировать тип перехватываемого трафика, TCP или UDP. По умолчанию отображается трафик обоих типов;

14) srcport порт: То же, что и предыдущая комбинация, только перехватывается трафик с заданным исходным портом;

15) less длина: Отображает пакеты с длиной, меньшей или равной заданной. Допустима также комбинация len<= длина;

16) greater длина: То же, что и предыдущая комбинация, только перехватывается трафик с длиной пакетов больше или равной указанной;

17) ipproto протокол: Перехватывает трафик заданного протокола. Допустимыми протоколами служат icmp, icmpv6, igmp, igrp, pim, ah, esp, vrrp, udp и tcp. Имена tcp, udp и icmp должны помещаться между двумя обратными косыми чертами, чтобы они не читались как ключевые слова. Пример: ipproto \tcp\;

18) ip6 proto протокол: Аналогично предыдущей комбинация, но для пакетов и типов IPv6;

19) ip6 protochain протокол: Ищет пакеты IPv6, имеющие заголовок указанного протокола;

20) ipprotochain протокол: То же, что и выше, но для пакетов IPv4;

21) ipbroadcastЖ Идентифицирует только широковещательный трафик, то есть трафик, имеющий все нули или все единицы в поле целевого адреса;

22) ethermulticastЖ Регистрирует вещательные пакеты Ethernet;

23) ipmulticast: Регистрирует вещательные пакеты IP;

24) ip6 multicast: Регистрирует вещательные пакеты IPv6;

25) etherproto протокол: Отображает трафик, который имеет указанный тип протокола Ethernet. Допустимыми именами протоколов служат ip, ipv6, arp, rarp, atalk, aarp, decnet, sca, lat, mopdl, moprc, iso, stp, ipx и netbeui. Эти имена являются также идентификаторами, поэтому они должны быть экранированы с помощью обратных косых черт;

26) decnetsrc хост: Перехватывает трафик DECnet с исходным адресом хоста;

27) decnetdst хост: Аналогична предыдущей комбинация, но фильтрует целевой адрес хоста;

28) decnet хост: Фильтрует трафик DECnet с исходным или целевым адресом хоста;

29) ip: Сокращенный вариант описанной выше комбинации etherprotoip. Перехватывает трафик, соответствующий Ethernet-протоколу ip;

30) ip6: Сокращенный вариант описанной выше комбинации etherprotoip6. Перехватывает трафик, соответствующий Ethernet-протоколу ip6;

31) arp: Сокращенный вариант описанной выше комбинации etherprotoarp. Перехватывает трафик, соответствующий Ethernet-протоколу arp;

32) rarp: Сокращенный вариант описанной выше комбинации etherprotorarp. Перехватывает трафик, соответствующий Ethernet-протоколу rarp;

33) atalk: Сокращенный вариант описанной выше комбинации etherprotoatalk. Перехватывает трафик, соответствующий Ethernet-протоколу atalk;

34) aarp: Сокращенный вариант описанной выше комбинации etherprotoaarp. Перехватывает трафик, соответствующий Ethernet-протоколу aarp;

35) decnet: Сокращенный вариант описанной выше комбинации etherprotodecnet. Перехватывает трафик, соответствующий Ethernet-протоколу decnet

36) iso: Сокращенный вариант описанной выше комбинации etherprotoiso. Перехватывает трафик, соответствующий Ethernet-протоколу iso;

37) stp: Сокращенный вариант описанной выше комбинации etherprotostp. Перехватывает трафик, соответствующий Ethernet-протоколу stp;

38) ipx: Сокращенный вариант описанной выше комбинации etherprotoipx. Перехватывает трафик, соответствующий Ethernet-протоколу ipx;

39) netbeui: Сокращенный вариант описанной выше комбинации etherprotonetbeui. Перехватывает трафик, соответствующий Ethernet-протоколу netbeui;

40) vlan:идентификатор_ВЛВС Перехватывает пакеты на основе стандарта 802.1QVLAN. Идентификатор виртуальной локальной сети можно опускать;

41) tcp: Сокращенная форма комбинации ipprototcp;

42) udp: Сокращенная форма комбинации ipprotoudp;

43) icmp: Сокращенная форма комбинации ipprotoicmp;

44) isoproto: протокол Перехватывает пакеты BOC с заданным типом протокола - clnp, esis или isis;

45) clnp: Сокращенная форма описанной выше комбинации с clnp в качестве протокола;

46) esis: Сокращенная форма комбинации isoproto протокол с esis в качестве протокола;

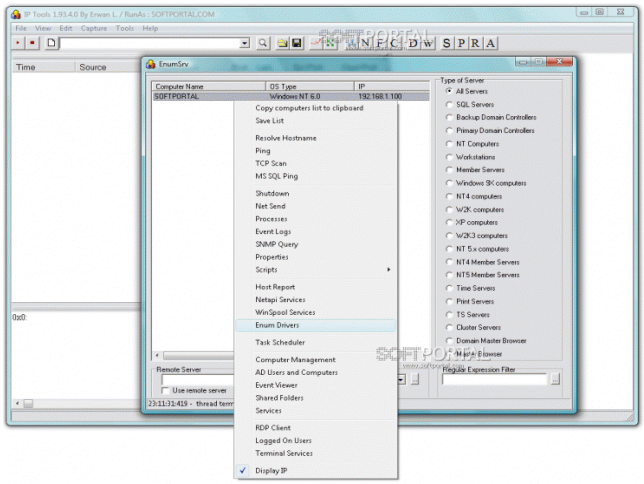
47) isis: Сокращенная форма комбинации isoproto протокол с isis в качестве протокола.

Анализатор IP Sniffer

IP Sniffer — это бесплатный сниффер для windows, который не требует установки и прав администратора. Несмотря на то, что IP Sniffer полностью бесплатен, он умеет не только наблюдать за потоком трафика, но и декодировать данные. Например, пароли от icq, skype, почты, браузерные пароли и т. п. При этом сниффер обладает простым графическим интерфейсом, который интуитивно понятен. Анализатор имеет основные функции, такие как фильтр, декодирование, воспроизведение, разбор...

IP Sniffer содержит ряд встроенных сетевых утилит:

- монитор трафика;
- отображение диаграмм чаще всего используемых IP адресов и протоколов;
- ARP (просмотр/удаление записей, отправка ответа);
- Netstat (отображение сетевых подключений с возможностью принудительно завершать выбранные);
- получение подробной информации о используемом сетевом адаптере;
- Spoofing (TCP, UDP, ICMP, ARP протоколов);
- WINS;
- DNS Query (using win32 DNSAPI);
- поиск DHCP серверов;
- служба WhoIs;
- преобразование IP в Hostname и наоборот;
- PING (хостов или подсети);
- сканирование определенного хоста или всей подсети на наличие открытых портов и т.д.



Для работы программы IP Sniffer может понадобиться установить утилиту WinPcap.

Отметим другие интересные возможности IP Sniffer:

- Ведет подробный монитор трафика.
- Даёт возможность удалять или просматривать записи, а также отправлять ответы.

- IP адреса или протоколы, которые используются чаще других можно вывести в виде диаграмм, что бы было проще анализировать траффик.
- Есть возможность получить Netbios имени, зная IP адрес.
- Поддерживает spoofing'a по UDP, ARP, TCP, ICMP протоколам.
- Можно получить информацию о сетевом адаптере, который используется.
- Поддерживает DHCP сервера, DNS Query, WINS.
- Поддерживает команды whois, ping и т.п.
- Умеет преобразовывать Hostname в ip и наоборот.
- Может сканировать ip адреса на наличие открытых портов или целой подсети.

Анализатор Wireshark

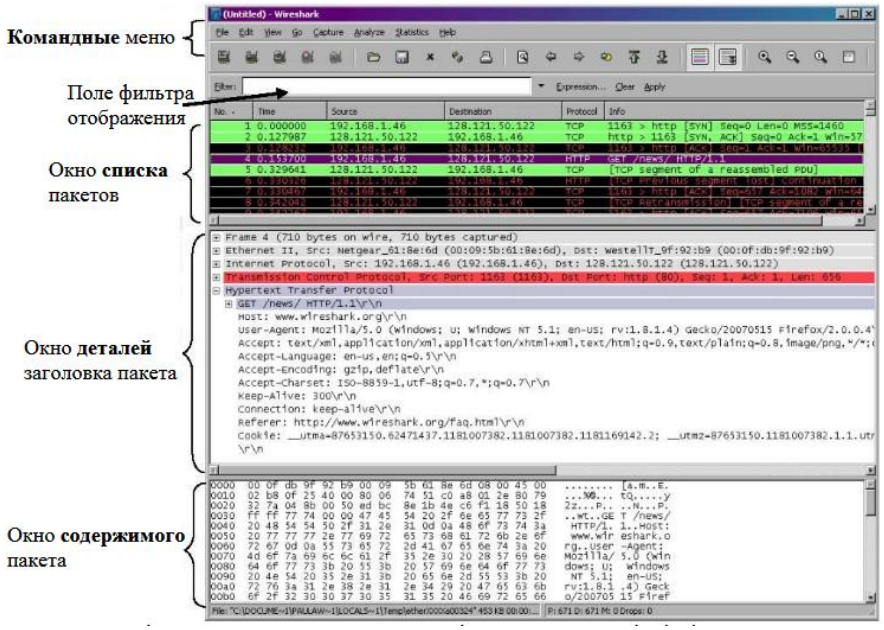
Анализатор Wireshark (ранее - Ethereal) - это программа-анализатор трафика для компьютерных проводных и беспроводных сетей, телефонных сетей и других. Программа позволяет пользователю просматривать весь проходящий по сети трафик в режиме реального времени.

Wireshark - это приложение, которое "знает" структуру различных сетевых протоколов, и поэтому позволяет разобрать сетевой пакет, отображая значение каждого поля протокола любого уровня. Wireshark умеет работать с множеством форматов входных данных, соответственно, можно открывать файлы данных, захваченных другими программами, что расширяет возможности захвата.

Анализатор захватывает пакеты, попадающие на интерфейс компьютера, сохраняет и отображает информацию о каждом пакете: тип протокола, структуру полей, адрес источника и адрес назначения, а также другие параметры. Он обеспечивает возможность сбора всех пакетов заданного соединения TCP и представления данных в удобном для просмотра виде.

Для активизации режима перехвата сетевых пакетов, надо выбрать пункт главного меню Capture -> Options и выбрать требуемые интерфейсы, на которых предполагается захватывать пакеты. Также следует при необходимости отключить или включить режим Promiscuous. С помощью кнопки "Options" также возможна установка нужных пользователю параметров работы программы.

Для запуска процедуры захвата, надо нажать кнопку "Start", после чего интерфейс программы будет иметь вид:



Основное окно Wireshark содержит три окна просмотра с различными уровнями детализации:

1. Верхнее окно содержит список всех собранных пакетов с кратким их описанием.

2. В среднем окне показывается дерево протоколов, инкапсулированных в кадр. При этом ветви дерева могут быть раскрыты для повышения уровня детализации выбранного протокола.

3. Нижнее окно содержит дамп пакета в шестнадцатеричном и текстовом представлении.

В верхнем окне по умолчанию 6 колонок: номера пакета в списке, временная метка, адреса и номера портов отправителя и получателя, тип протокола и краткое описание пакета. Выбрав в списке верхнего окна пакет, его содержимое можно просмотреть в среднем окне). В среднем окне отображается дерево всех протоколов инкапсулированных в пакет, который был выбран в верхнем окне. Дерево отображает каждое поле и его значение для заголовков всех протоколов стека.

Если в среднем окне кликнуть мышкой на пиктограмме со знаком плюс у заголовка какого-либо протокола, то можно просмотреть состав и содержимое этого протокола. При этом выбранная область протокола выделяется подсветкой в дампе пакета нижнего окна.

При захвате большого количества пакетов, работа со списком верхнего окна будет не очень комфортной. Для удобства работы в программе предусмотрена возможность сортировки списка пакетов по любому полю, а также его фильтрация по сложным критериям.

Меню "Filter" позволяет инициировать диалог построения фильтра. Справа от этой кнопки - поле для ввода и редактирования фильтра. Оно служит для отображения текущего фильтра и просмотра в режиме прокрутки списка ранее использованных фильтров. Кнопка "Clear" очищает окно ввода фильтра, а кнопка "Apply" инициирует выбранный фильтр.

Настройка параметров процедуры захвата

Запуск программы возможен и при установленных по умолчанию параметрах режима захвата пакетов. Для изменения и выбора параметров следует в основном меню выбрать Capture -> Options. Рассмотрим некоторые наиболее часто используемые параметры:

Поле "Interface:" позволяет указать интерфейс, на котором требуется анализировать пакеты. Можно задать только один интерфейс, причем только из числа тех, которые обнаружены анализатором.

Поле "Enable promiscuous mode" переводит интерфейс анализатора в режим "неразборчивого" захвата. Если этот режим не включен, анализатор будет захватывать только пакеты, адресованные в компьютер, на котором установлен анализатор, или исходящие из него.

Можно также задать максимальный объем данных, число пакетов, всемя захвата.

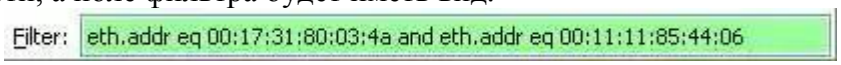
Кнопка и поле "Capture Filter" позволяют задать фильтр пакетов. Пустое значение означает захват без фильтрации и. Если щелкнуть кнопку "Filter", то появится диалоговое окно для построения или выбора готового фильтра.

Меню "Output" служит для ввода имени файла, в котором будут сохранены результаты захвата.

Программа позволяет контролировать как любые пакеты, так и пакеты, только конкретного протокола, циркулирующего в сети. Например, пакеты, проходящие, по протоколу HTTP. После этого для удобства проведем сортировку списка пакетов по протоколам, кликнув на заголовок колонки Protocol. Далее, находим пакет протокола HTTP и, нажав правую кнопку мыши, в контекстном меню выбираем Conversation filter>IP, тем самым, выбрав фильтрацию списка пакетов только данного перечня адресов отправителя и получателя.

Для фильтрации пакетов по MAC-адресам источника и приемника выберем пакет протокола HTTP для интересующих нас узлов и, нажав правую кнопку мыши, в контекстном меню выберем Conversation filter>Ethernet. После этого в верхнем окне

останутся только пакеты протокола HTTP, которыми обменивались интересующие нас узлы сети, а поле фильтра будет иметь вид.



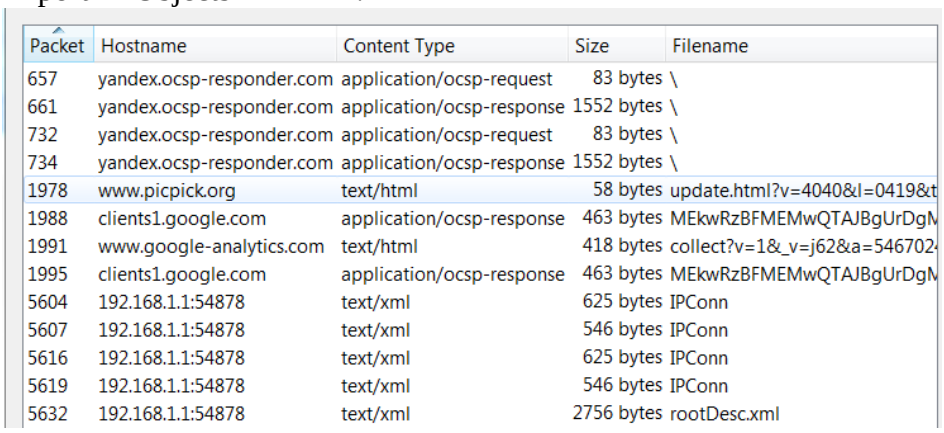
С помощью системы поиска можно проверить пакеты, соответствующие адресации, установленной при фильтрации. После того как нужный пакет будет найден в списке захваченных пакетов, в дереве протоколов можно в ветви Line-based text data найти значения полей Login, Domain, Password.

Возможности программы Wireshark

В анализаторе протоколов Wireshark предусмотрена возможность маркировки пакета при помощи установки курсора на нужный пакет и выбора в контекстном меню после нажатия правой кнопки мыши Mark Packet. Далее возможен быстрый поиск маркированного пакета при помощи главного меню Edit -> Find Next Mark.

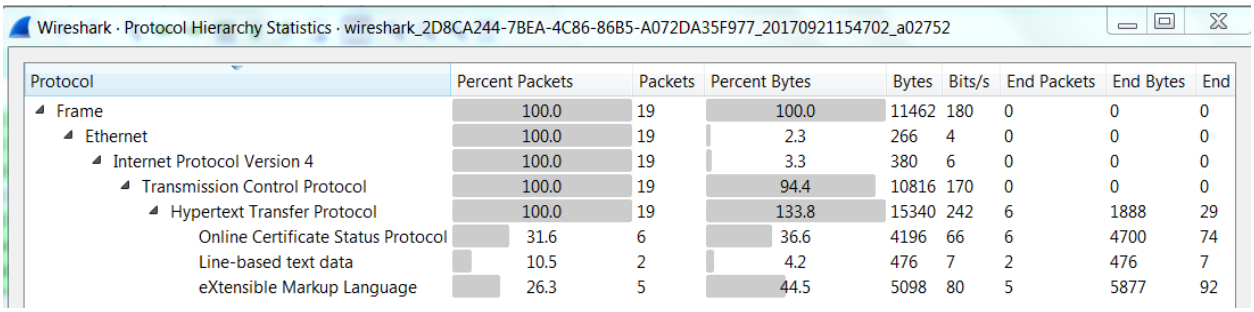
Практически важной представляется предусмотренная в программе возможность экспорта суммарной информации о пакетах и дереве протоколов в файл формата *.txt при помощи главного меню File -> Export -> File.

Кроме этого, 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.



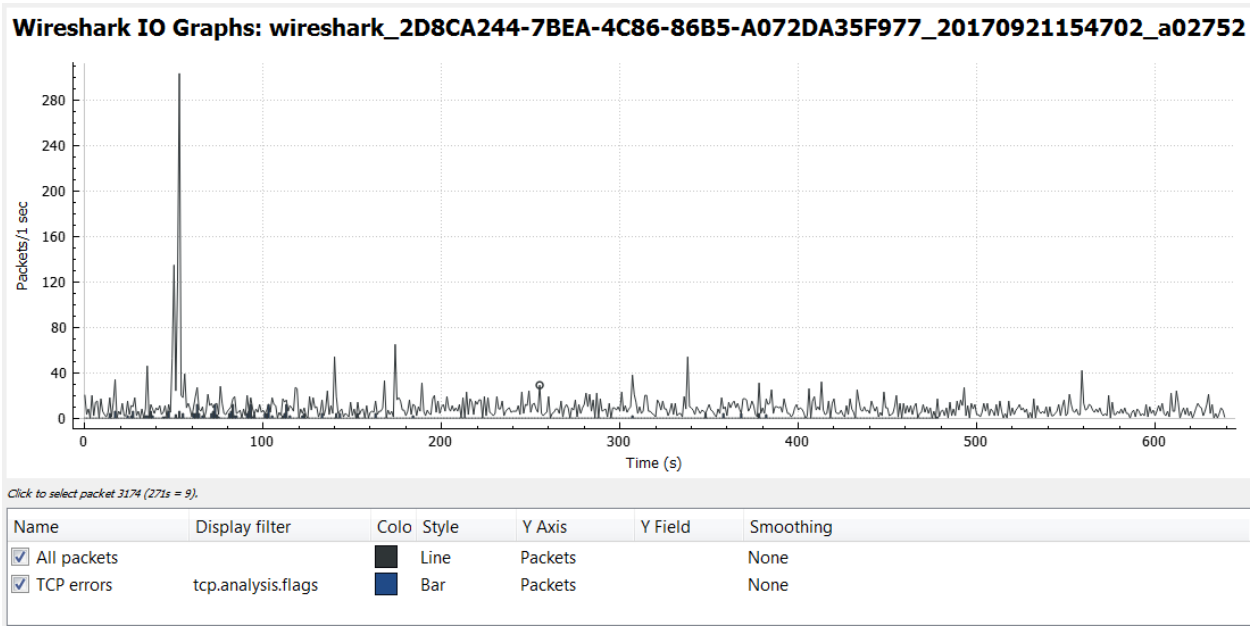
Packet	Hostname	Content Type	Size	Filename
657	yandex.ocsp-responder.com	application/ocsp-request	83 bytes	\
661	yandex.ocsp-responder.com	application/ocsp-response	1552 bytes	\
732	yandex.ocsp-responder.com	application/ocsp-request	83 bytes	\
734	yandex.ocsp-responder.com	application/ocsp-response	1552 bytes	\
1978	www.picpick.org	text/html	58 bytes	update.html?v=4040&l=0419&t
1988	clients1.google.com	application/ocsp-response	463 bytes	MEkwRzBFMEMwQTAJBgUrDgM
1991	www.google-analytics.com	text/html	418 bytes	collect?v=1&_v=j62&a=546702
1995	clients1.google.com	application/ocsp-response	463 bytes	MEkwRzBFMEMwQTAJBgUrDgM
5604	192.168.1.1:54878	text/xml	625 bytes	IPConn
5607	192.168.1.1:54878	text/xml	546 bytes	IPConn
5616	192.168.1.1:54878	text/xml	625 bytes	IPConn
5619	192.168.1.1:54878	text/xml	546 bytes	IPConn
5632	192.168.1.1:54878	text/xml	2756 bytes	rootDesc.xml

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:



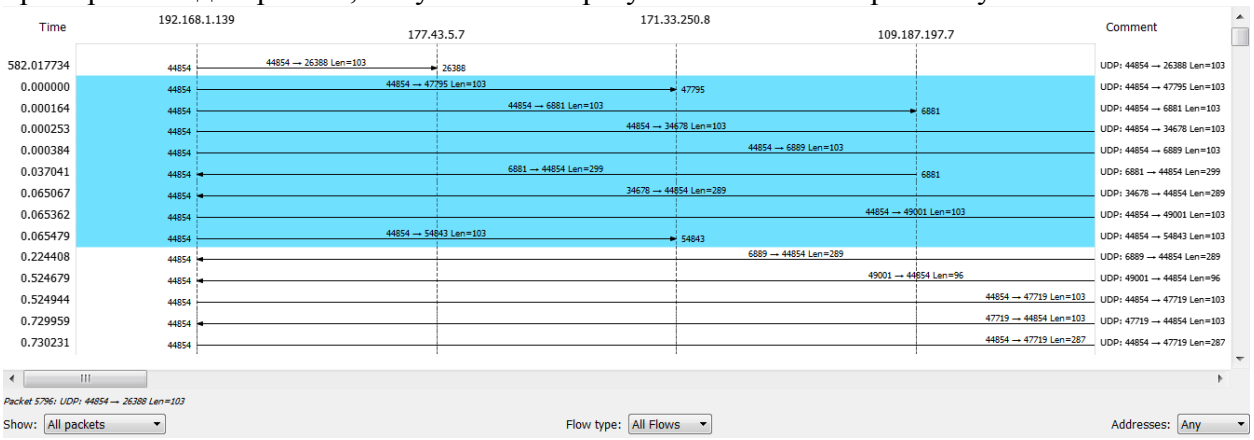
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End
Frame	100.0	19	100.0	11462	180	0	0	0
Ethernet	100.0	19	2.3	266	4	0	0	0
Internet Protocol Version 4	100.0	19	3.3	380	6	0	0	0
Transmission Control Protocol	100.0	19	94.4	10816	170	0	0	0
Hypertext Transfer Protocol	100.0	19	133.8	15340	242	6	1888	29
Online Certificate Status Protocol	31.6	6	36.6	4196	66	6	4700	74
Line-based text data	10.5	2	4.2	476	7	2	476	7
eXtensible Markup Language	26.3	5	44.5	5098	80	5	5877	92

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.



Анализ пакетов

Программа предоставляет пользователю набор средств автоматизированного анализа полученной информации, среди которых инструмент Flow Graph (меню Statistics). Он позволяет построить диаграмму перемещения пакетов по узлам. На рис.22 представлен пример такой диаграммы, полученный по результатам анализа работы утилиты traceroute.



Обычно для выполнения анализа трафика следует выполнить следующие действия.

1.Открыть окно настройки параметров анализа;

2.Выбрать требуемый сетевой интерфейс из списка обнаруженных;

3.Ввести необходимые условия фильтрации.

4.Если требуется, выбрать режим автоматического сохранения информации, условия останова работы анализатора;

5.Установить флаги Update list of packets in real time и Automatic scrolling in live capture;

6.Нажать кнопку Start.

7.Отслеживать обрабатываемый трафик в основном окне.

8.При необходимости остановить анализ, сменить параметры фильтрации и продолжить работу.

9.Остановить анализ, когда вся необходимая информация представлена на экране.

10.Анализировать информацию, просматривая содержимое пакетов и используя встроенные средства анализа.

Фильтрация пакетов в процессе захвата.

Ранее уже говорилось про возможность фильтрации списка захваченных пакетов. При этом фильтр формировался по образцу, когда мы выбирали тот или иной пакет и использовали соответствующую опцию меню.

Но еще более важной функцией Wireshark является возможность фильтрации пакетов в процессе их захвата. Для этих целей необходимо правильно сформировать необходимые логические выражения и использовать его в качестве фильтров.

Для записи фильтра используется последовательность простых выражений (примитивов), соединенных логическими связками and или or, перед которыми может присутствовать отрицание not:

[not] <примитив> [and | or [not] <примитив> ...]

Например, чтобы захватить трафик, адресованный к узлу с IP-адресом 192.168.12.14 или исходящий из него, используется фильтр

tcp port 23 and host 192.168.12.14

Если нужно захватить весь трафик, кроме трафика, адресованного узлу 192.168.12.14 или исходящего из него, используется фильтр

tcp port 23 and not host 192.168.12.14

Следует отметить возможность построения и более сложных конструкций логических выражений при вызове окна диалога построения фильтра.

В последних версиях Wireshark для формирования фильтров используется объектная модель примитивов. Это приводит к тому, что, при ручном формировании логических выражений введя тот или иной объект примитива и нажав точку (например, ip.) появляется выпадающий список со свойствами этого примитива. Выбрав в списке нужное свойство, оно автоматически вставляется в логическое выражение фильтра.

Порядок выполнения работы

Задание № 1. Знакомство с работой анализатора

1. Запустите анализатор в режиме захвата всех пакетов, проходящих по сети в течении нескольких секунд. Количество захватываемых пакетов ограничить 20. Результаты сохранить в файл отчета.
2. Запустите браузер и обратитесь к удаленному сайту согласно варианту заданий.
3. Определите IP адрес удаленного узла, маршрут доступа к нему и сохраните результаты для отчета.
4. Настройте анализатор для захвата пакетов от данного IP-адреса.
5. Выведите для отчета таблицу иерархии протоколов при обращении к данному узлу.
6. Выведите для отчета график передачи пакетов в единицу времени при при обращении к данному узлу.
7. Постройте диаграмму перемещения пакетов по узлам маршрута при обращении к данному узлу.
8. Запустите анализатор в режиме перехвата широковещательного трафика в течении нескольких секунд. Количество захватываемых пакетов ограничить 20. Результаты сохраните для отчета.
9. Выполните наблюдение за входящим и исходящим трафиком 23 порта. Выполните просмотр входящего и исходящего трафика заданного в варианте хоста за исключением трафика 23 порта.

Задание 2. Захват и анализ пакетов в локальной сети

1. Произвести захват и анализ пакетов, передающихся через сеть при авторизации компьютера в локальной сети.

2. Произвести эхо запрос к соседнему узлу локальной сети и зафиксируйте ответ. Проведите его анализ и сохраните результаты для отчета.

3. Скопируйте с соседнего узла файл. Зафиксируйте и проанализируйте возникающий трафик.

4. Внесите результаты анализа в отчет.

Задание 3. Анализ IP трафика в глобальной сети

1. Запустите анализатор и подключитесь к серверу, согласно варианту заданий.
2. Перехватите и сохраните в файл 50 кадров при обращении к сайту согласно варианту заданий.
3. Проанализируйте IP пакеты. Согласно описанию формата заголовка IP пакета заполните бланк аналитического отчета. Примечание. Наличие или отсутствие поля «Опции» в заголовке, а так же его длину, можно выяснить, лишь проанализировав значения других полей.

Задание 4. Анализ ICMP трафика в глобальной сети

1. Запустите анализатор так, чтобы он перехватывал только пакеты только протокола ICMP, отправленные на определенный по варианту заданий IP-адрес. Для генерирования пакетов воспользоваться утилитой ping.
2. В соответствии с вариантом пошлите ICMP эхо-запрос на удалённый хост и определить количество «хопов» до удалённого хоста. Сохраните результаты для отчета. Количество пакетов ограничить 10..
3. Проанализируйте ответные пакеты ICMP трафика. Внесите результаты анализа в отчет.

Задание 5. Сканирование портов на узлах глобальной сети

1. Сделайте запросы к удаленному хосту согласно варианту.

2. Просканируйте порты удалённого хоста. Зафиксируйте и опишите службы на открытых портах.

3. Объясните реакцию хоста на каждый запрос.

4. Внесите полученные результаты анализа в отчет.

Защита работа

Для защиты данной работы, необходимо:

1. Предоставить отчеты об обмене узлов IP датаграммами с распечатками полученных текстовых файлов с пояснениями, что в них содержится.

2. Предоставить заполненный аналитический отчет с пояснениями по полям заголовка IP пакета в таблице.

3. Продемонстрировать результат преподавателю и ответить на контрольные вопросы.

Содержание отчета

1. Протоколы выполнения работы в виде сохраненных файлов захвата пакетов с текстовым пояснением их содержания.

2. Распечатки скриншотов с результатами выполнения заданий работы.

3. Результаты анализа пакетов, описание служб, протоколов, формата пакета.

Варианты заданий на лабораторную работу

Вариант	Удаленный хост	Вариант	Удаленный хост
1	президент.рф	**21**	Yandex.ru
2	правительство.рф	**22**	Microsoft.com
3	яндекс.рф	**23**	Google.com
4	гугл.рф	**24**	fast-bit.org
5	известия.рф	**25**	www.sciencedirect.com
6	кремль.рф	**26**	kias.rfbr.ru
7	лента.рф	**27**	elibrary.ru
8	жж.рф	**28**	vk.com
9	вконтакте.рф	**29**	mail.ru
10	ульяновск.рф	**30**	lenta.ru
11	псков.рф	**31**	asus.com
12	ростов-на-дону.рф	**32**	youtube.com
13	самара.рф	**33**	orcid.org
14	росавтодор.рф	**34**	rzd.ru
15	роспатент.рф	**35**	grant.rscf.ru
16	лаборатория касперского.рф	**36**	nalog.ru
17	консультант-плюс.рф	**37**	rae.ru
18	сбербанк.рф	**38**	ieee.org
19	найдется-все.рф	**39**	researcherid.com
20	вим-авиа.рф	**40**	ok.ru