Лабораторная работа 5.
«Анализ трафика компьютерных сетей утилитой Wireshark»

1 Цель и краткая характеристика работы

Цель работы – изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

В процессе выполнения домашнего задания выполняются наблюдения за передаваемым трафиком с компьютера пользователя в Интернет и в обратном направлении. Применение специализированной утилиты Wireshark позволяет наблюдать структуру передаваемых кадров, пакетов и сегментов данных различных сетевых протоколов. При выполнении УИР требуется анализировать последовательности команд и назначение служебных данных, используемых для организации обмена данными в следующих протоколах: ARP, DNS, FTP, HTTP, DHCP.

2 Теоретическая справка

Процесс передачи данных по компьютерным сетям является сложным комплексом процедур, выполняемых с применением большого количества разнообразных программных и аппаратных средств. Для упрощения анализа и проектирования таких сложных систем, общепринятой практикой является декомпозиция сложного процесса на модули и/или иерархические структуры.

Обычно целью декомпозиции является получение таких модулей, которые выполняют отведённые им функции изолированно от других модулей, передавая соседним модулям лишь конечные результаты работы. Это позволяет рассматривать и проектировать модули независимо друг от друга различным, не связанным друг с другом группам инженеров, каждая из которых обладает узкой квалификацией, необходимой для реализации конкретного модуля. Кроме этого, система, обладающая модульной структурой, позволяет при необходимости модифицировать внутреннюю реализацию отдельных модулей, не изменяя что-либо в соседних модулях.

В компьютерных сетях общепринятой моделью декомпозиции процесса передачи данных является OSI-модель, разработанная международной организацией по стандартизации. OSI означает “Open Systems Interconnection”, т.е. взаимодействие открытых систем. Модель OSI определяет 7 модулей, называемых уровнями (layer), каждый из которых описывает реализацию некоторого множества родственных сетевых операций, которые выполняет ЭВМ, начиная от момента получения данных от пользователя и заканчивая

отправлением физического сигнала (например, радиоволны) в сеть. Уровни связываются между собой строго последовательно:

“пользователь” ⇔ 7 ⇔ 6 ⇔ 5 ⇔ 4 ⇔ 3 ⇔ 2 ⇔ 1 ⇔ “сеть”

(здесь цифрами обозначены номера соответствующих уровней). Это значит, что после выполнения сетевых функций некоторого уровня результаты его деятельности могут быть переданы только соседним уровням. Результатами деятельности являются закодированные блоки данных, называемые PDU (протокольные блоки данных). Обычно при движении PDU по OSI-модели от “пользователя” в “сеть” каждый уровень дополняет полученный PDU своими служебными данными. В результате PDU 2-го уровня может иметь следующую структуру (прямоугольники обозначают последовательность бит: количество бит пропорционально длине прямоугольника):

СД2	СД3	СД4	СД5	СД6	СД7	ДП

Здесь СД*i* – это служебные данные, добавленные *i*-м уровнем, а ДП – это данные пользователя, который он хотел передать по сети. Служебные данные некоторых уровней могут быть организованы в виде двух частей: заголовка и концевика (в этом случае структура PDU выглядит иначе, чем показано выше).

При движении PDU по OSI-модели в обратном направлении (т.е. из “сети” к “пользователю”) каждый уровень сначала использует одноимённые служебные данные для выполнения заданной сетевой функции, а затем “отстёгивает” их при передаче следующему в цепочке уровню. Описанный подход гарантирует невмешательство уровней в работу друг друга, обеспечивая хорошую “модульность“ процесса, однако предполагает достаточно большое количество служебных данных, которые могут дублироваться на разных уровнях, что увеличивает накладные расходы на передачу полезных ДП.

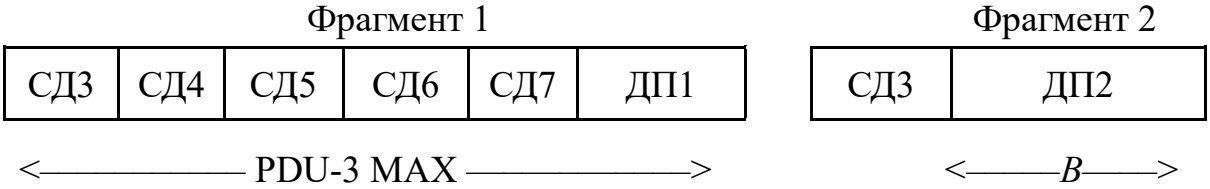
В некоторых случаях правила уровня могут накладывать ограничения на размер PDU, который может быть корректно обработан уровнем. В этом случае при попытке передать PDU большего размера, PDU будет либо отвергнут с сообщением об ошибке, либо будет фрагментирован на несколько частей, каждая из которых будет передана независимо. При использовании фрагментирования требуется, чтобы принимающая фрагменты сторона могла соединить фрагменты воедино.

Пусть следующий PDU 4-го уровня имеет размер, который превышает предельно допустимый PDU 3-го уровня (PDU-3 MAX) на *B* байт:

СД4	СД5	СД6	СД7	ДП

<——————— PDU-3 MAX ———————> <———*B*———>

В этом случае на вход 2-го уровня вместо одного будет передано сразу несколько PDU 3-го уровня, которые будут представлять собой фрагменты, имеющие допустимую для 3-го уровня длину:



В рассмотренном случае удалось разбить исходный PDU 4-го уровня всего на два PDU 3-го уровня. Данные пользователя пришлось “разрезать” на две части ДП1 и ДП2. Суммарный размер ДП1 и ДП2, очевидно, равен ДП. Однако при фрагментировании пришлось добавить СД3 в оба фрагмента, что привело к увеличению доли накладных расходов в передаваемом сообщении. Это необходимо для того, чтобы была возможность корректно собрать PDU-4 из фрагментов на приёмной стороне.

Дадим краткую характеристику каждому из 7 уровней OSI-модели, двигаясь по модели в направлении от пользователя в сеть (более подробное описание уровней см. в [1]).

Прикладной уровень (Application Layer, L7) описывает, как выглядит процесс передачи с точки зрения конечного пользователя или приложения. L7 предоставляет понятные пользователю высокоуровневые “рычаги” для получения сервисов уровней L1-L6. Конечный пользователь или приложение при работе с сетью взаимодействует только с L7, а все нижележащие уровни от него скрыты, т.е. инкапсулированы в L7. Этот уровень имеет нечёткие границы, так как может описывать не только функции сетевого приложения, но и возможные действия пользователя. На уровне L7 *может* описываться:

- авторизация и аутентификация пользователя;
- контроль целостности конечных пользовательских данных, которые были получены из сети;
- синхронизация действий или файлов пользователей, взаимодействующих по сети (например, при совместном редактировании файла несколькими пользователями);

- … а также любая из функций уровней L2-L6 (см. ниже), если она не была реализована на L2-L6, либо была реализована не в полном соответствии с потребностями приложения или пользователя.

Уровень представления (Presentation Layer, L6) описывает, как взаимодействующие стороны “договариваются” о формате, в котором будут представлены данные пользователя при передаче по сети. На этом уровне могут описываться:

- процедура согласования формата представления данных на этапе установки соединения (например, выбор UTF-8 для кодирования текста пользователя или выбор алгоритма сжатия и его параметров и т.п.);
- правила изменения текущего формата представления данных в некотором уже установленном соединении (например, изменение кодека, сжимающего аудиопоток во время VoIP-разговора, при обнаружении перегрузки канала связи);
- описание синтаксиса выбранного формата представления данных, если он не является общеизвестным стандартом и не может быть описан простой ссылкой на стандарт (например, новый патентованный алгоритм сжатия или особый порядок следования байтов, отличный от Big Endian и Little Endian).

Сеансовый уровень (Session Layer, L5) описывает процесс установки, разрыва и поддержания соединений. На этом уровне может описываться:

- процедура установки соединения и согласования параметров соединения (например, требований QoS – см. ниже), при этом фактическая реализация запрошенных требований осуществляется на уровне L4;
- процедура разрыва соединения как при явном запросе пользователя, так при получении от L4 сообщения о невозможности выполнить запрошенные требования QoS (может быть описан штатный разрыв без потери данных пользователя и/или быстрый “жёсткий” сброс соединения с риском потери данных);
- процедура (ре)синхронизации состояния соединения (соединение может рассинхронизироваться при возникновении ошибок в сети, при переводе часов, при выходе за границу окна передачи и т.п.).

Транспортный уровень (Transport Layer, L4) описывает процесс межоконечной (end-to-end, “из конца в конец”) передачи данных по сети, т.е.

передачу с точки зрения наблюдателя, для которого все промежуточные сетевые устройства между абонентами рассматриваются как единый “черный ящик”, структура которого неизвестна. На этом уровне может описываться:

- процедура установки/поддержания/разрыва соединения и передачи данных с учётом соблюдения требований QoS, полученных от L5 (например, может понадобиться установить сразу несколько L4-соединений или выбрать такие L4-параметры, которые гарантируют соблюдение QoS-требований с большим запасом, ввиду отсутствия возможности точной настройки);
- процедура реагирования на обнаружение искажённых или потерянных пакетов (следует ли повторить передачу или же допустимо игнорировать потерю/искажение?);
- процедура сохранения корректного порядка поступления PDU конечному абоненту (PDU снабжаются порядковыми номерами, при этом может потребоваться буфер для временного хранения PDU, поступивших с нарушением порядка);
- процедуры манипуляции с размером PDU: мультиплексирование потоков, разбиение больших PDU на более мелкие, объединение маленьких PDU в большие и т.п.

Сетевой уровень (Network Layer, L3) описывает процесс передачи PDU через промежуточные узлы сети, включая выбор маршрута следования при наличии нескольких маршрутов. При этом маршрут передачи может пересекать несколько объединённых разнотипных сетей. Сетевой уровень полностью скрывает от вышестоящих уровней (L4-L7) особенности маршрутизации и передачи PDU через разнородные сети, т.к. реализует их самостоятельно или средствами уровня L2. На уровне L3 может описываться:

- установка/поддержание/разрыв соединения в условиях, когда необходимо пересекать границы нескольких сетей (при пересечении границы между сетями разных провайдеров может потребоваться инициировать установку соединения независимо от установки межоконечного соединения на L5);
- процедура сохранения корректного порядка поступления PDU на границе сетей (ср. с аналогичным пунктом в L3);
- правила маршрутизации и построения маршрутных таблиц при пересечении границ сетей;

- процедуры манипуляции с размером PDU (см. L4);
- правила подтверждения получения PDU приёмником.

Канальный уровень (Data Link Layer, L2) описывает логические правила передачи PDU в пределах простой сети, построенной в рамках единой технологии с одинаковыми однотипными линиями связи. L2 скрывает от вышестоящих уровней физические особенности сети. На этом уровне может описываться:

- установка, поддержание и разрыв соединения в рамках одной локальной сети с возможностью согласования узлами параметров передачи (при подключении к Wi-Fi требуется установить L2-соединение с базовой станцией Wi-Fi независимо от установки соединения на L3 и L5);
- процедура реагирования на обнаружение искажённых или потерянных пакетов (сравни с L4);
- синхронизация приемо-передатчиков сетевых устройств для корректного распознавания границ PDU (например, отправка блока известной абонентам длины с чередованием 0 и 1: “01010101...”; или использование запрещённых сигналов J, K);
- процедура сохранения корректного порядка поступления PDU внутри сети (сравни с аналогичным пунктом в L3 и L4);
- правила разделения потока PDU на несколько подпотоков для возможности их одновременной передачи по нескольким физическим линиям связи (например, с использованием нескольких радиоканалов с разным диапазоном частот);
- правила маршрутизации и построения маршрутных таблиц внутри сети (сравни с аналогичным пунктом в L3; обычно в L2-сетях маршрутизация не требуется, т.к. в них всегда существует ровно один маршрут).

Физический уровень (Physical Layer, L1) описывает с физической точки зрения процессы передачи PDU по некоторой конкретной линии связи. Сюда может входить спецификация физических свойств:

- *среды передачи*: ширина полосы пропускания радиоканала в МГц, максимальная длина провода при передаче по витой паре или оптоволокну, количество и назначение проводов или волокон в кабеле и т.п.;

- *передаваемого сигнала*: используемые длины волн или напряжение тока, способ кодирования битов в виде конкретного уровня напряжения, длительность времени передачи бита или группы битов, мультиплексирование нескольких физических сигналов в одной линии связи, показатели QoS линии связи (задержка распространения сигнала, доля битовых ошибок BER, скорость передачи в бодах и др.) и т.п.;
- *сетевого оборудования*: количество и назначение контактов в штекере сетевой карты или маршрутизатора, количество и физическое устройство антенн в радиопередатчике, способ передачи (полнодуплексный, полудуплексный, симплексный), способ активации линии связи при включении или начале передачи и т.п.

Существуют также некоторые универсальные функции, которые реализуются почти на всех уровнях OSI-модели. К таким функциям могут относиться следующие:

- **Адресация**. На каждом уровне (кроме L6) для идентификации взаимодействующих сторон может использоваться адрес определённого формата. На L2 это может быть адрес физического устройства сети, специфичный для некоторой конкретной технологии построения сети; на L3 это будет универсальный межсетевой адрес, пригодный для передачи между разнородными сетями; на L5 адресом будет номер соединения; на L7 адресом может быть имя пользователя или понятный человеку текстовый адрес. В итоге к моменту формирования PDU L1 в передаваемом блоке данных может содержаться 6 различных адресов-идентификаторов.
- **Обнаружение ошибок**. На каждом уровне есть своя специфика процесса обнаружения ошибок. На L1 об ошибке может сигнализировать запрещённое значение напряжения тока или неверная последовательность радиоимпульсов. На более высоких уровнях для обнаружения ошибок могут использоваться различные виды контрольных сумм, которые в том числе позволяют исправлять найденные ошибки (например, код Хэмминга). Если на L3 ошибки могут быть обнаружены в промежуточных узлах сети (при пересечении границ сетей), то на L4 проверка на ошибки возможна только в конечном узле-получателе. Ещё один класс обнаруживаемых ошибок связан с нарушением логических правил протокола обмена сообщениями, например, если абонент высылает первый блок данных до окончания процедуры установки соединения на L5.

- **Показатели QoS** (Quality of Service), т.е. метрики качества передачи данных по сети, например: задержка передачи, время установления соединения, доля потерянных пакетов и др. На разных уровнях OSI-модели рассматриваются различные аспекты QoS. На L1 метрики характеризуют качество передачи по одной конкретной линии связи, например в виде значения мощности сигнала базовой станции WiFi; на L2 контролируется QoS в рамках локальной сети; на L3 описывается QoS при пересечении границ сетей; на L4 контролируется качество передачи из конца в конец; на L5 описан порядок согласования абонентами желаемых показателей QoS во время установки соединения; на L7 пользователь может запросить желаемый битрейт потока для обеспечения нужного качества сетевой видеотрансляции и т.д.

- **Установка соединения**. При описании функций L2, L3 и L5 (см. выше) показано, что в процессе передачи данных может потребоваться выполнить несколько независимых процедур установки соединения. Например: L2-соедниение к Wi-Fi-точке, L3-соединение к провайдеру Интернет-услуг, L5-соединение или L7-содинение с публичным FTP-сервером, находящимся в Интернете (т.е. с конечным адресатом).

Существующие реализации OSI-модели. Стандарт с описанием OSI-модели был опубликован в 1984 году, однако с тех пор так и не появилось ни одной популярной сетевой технологии, которая бы строго реализовала все уровни этой модели. Наиболее популярный стек сетевых протоколов TCP/IP, разработанный до публикации OSI-модели, лишь с большой натяжкой можно соотнести с уровнями OSI:

- **Канальный уровень TCP/IP** приблизительно реализует функции L1 и L2. В качестве адреса на этом уровне используется MAC-адрес сетевого устройства, а PDU этого уровня называется кадром (фреймом, frame). Этот уровень описывает процесс передачи данных в рамках локальной сети.

- **Сетевой уровень TCP/IP** приблизительно соответствует L3. В качестве адреса на этом уровне используется IP-адрес, а PDU этого уровня называется пакетом (packet). Этот уровень описывает процесс передачи данных через несколько объединенных и, возможно, разнородных локальных сетей.

- **Транспортный уровень TCP/IP** приблизительно реализует функции L4 и L5. В качестве адреса на этом уровне используется пара чисел,

однозначно идентифицирующих соединение: порт отправителя и порт получателя (например, UDP-порты), а PDU этого уровня называется сегментом или датаграммой (segment, datagram).

- **Прикладной уровень TCP/IP** приблизительно реализует функции L5, L6 и L7 (обратите внимание, что L5 фигурирует дважды: на прикладном и на транспортном уровне TCP/IP). В качестве адреса на этом уровне используется URL сайта, DNS-имя хоста, имя пользователя, email-адрес и т.д.

3. Этапы выполнения работы и варианты заданий

Для выполнения УИР необходимо установить на компьютер бесплатно распространяемую программу Wireshark, представляющую из себя анализатор сетевых пакетов, проходящих через интерфейсы компьютера. Скачать Wireshark можно с официального сайта: https://www.wireshark.org/#download. На рисунке представлено главное окно *Wireshark.*

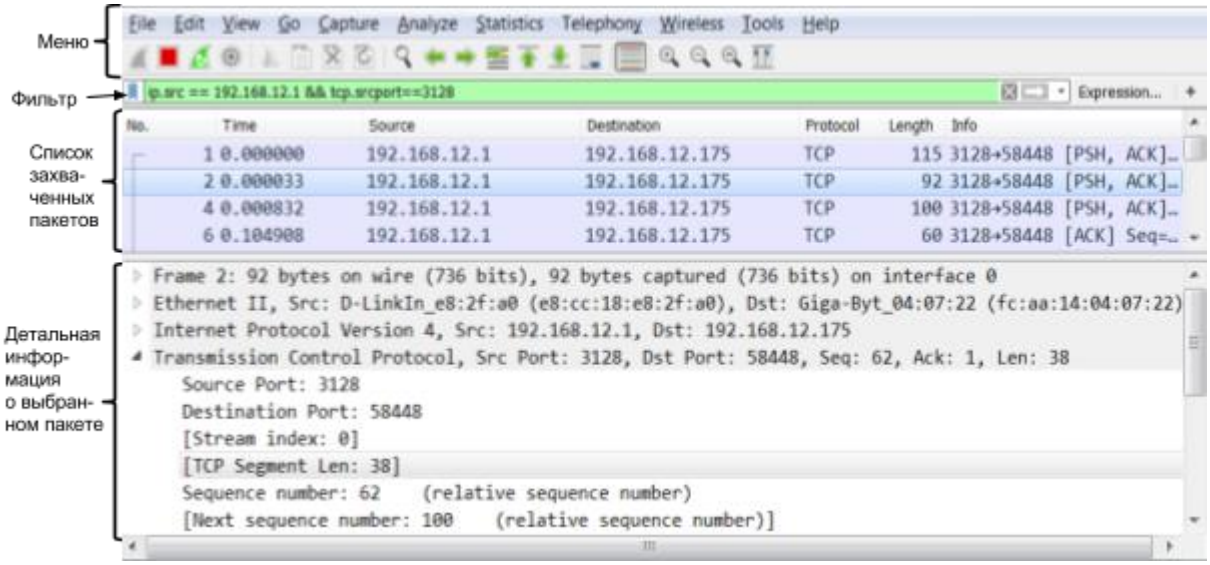


Рисунок. Графический интерфейс пользователя Wireshark

Используя “Меню”, можно выбрать сетевой интерфейс, “прослушивание” которого будет осуществлять Wireshark (кнопка “Capture options”). При выполнении работы следует удостовериться, что интерфейс выбран правильно, так как при наличии нескольких каналов доступа в Интернет (Wi-Fi, 4G, FastEthernet), как правило, по умолчанию используется только один из них и именно с него Wireshark должен “захватывать” проходящие пакеты.

В поле “Фильтр” пользователь может указать булево выражение (в стиле языка C), которое используется для выборочного отображения захваченных

пакетов в “Списке захваченных пакетов”. Например, если в “Фильтре” указать строку “(*ip.src==192.168.12.1) && (tcp.srcport==3128)*” (без кавычек), то в “Списке захваченных пакетов” будут отображаться только те пакеты, которые были отправлены с IP-адреса *192.168.12.1* и при этом в поле “порт источника” протокола TCP содержат число 3128. Если фильтр принимает значение “http”, то будут отображаться только пакеты, переданные с использованием протокола http.

Дальнейшее выполнение работы состоит из следующих шагов:

1. Запустить Wireshark (иногда для этого требуются права Администратора). В появившемся окне выбрать интерфейс, для которого необходимо осуществлять анализ проходящих через него пакетов. В качестве интерфейса, используемого для захвата трафика, выбрать физический адаптер, через который компьютер подключён к Интернету (обычно этот адаптер называется Local или “Подключение по локальной сети”). Если меню для выбора адаптера не появляется при запуске Wireshark, нужно запустить из “Меню” команду “Capture->Options”. После выбора адаптеру, нужно запустить процесс захвата трафика (кнопка Start).
2. Инициировать процесс передачи трафика по сети (например, в браузере открыть сайт, заданный по варианту, или запустить соответствующую сетевую утилиту – см. ниже);
3. Установить значение “Фильтра”, чтобы из всего множества перехватываемых пакетов Wireshark отобразил только те, которые имеют отношение к выполняемому заданию. Для корректного создания фильтра следует пользоваться всплывающими подсказками Wireshark, которые активизируются при наборе фильтра. В качестве альтернативного способа можно использовать интерактивный конструктор фильтра, нажав на кнопку “Expression” в правой части элемента “Фильтр”.
4. Дождаться появления данных в списке захваченных пакетов и убедиться, что количество пакетов достаточно для выполнения задания.
5. Сохранить захваченный трафик в файл-трассу (pcap). Указанный файл нужно предъявить по первому требованию преподавателя во время защиты, если в этом возникнет необходимость.
6. Описать в отчёте структуру наблюдаемых PDU (т.е. протокольных блоков данных: кадров, пакетов, сегментов) как для запросов, так и ответов. Указать название и назначение всех заголовков всех уровней OSI-модели в пакетах с учётом порядка инкапсуляции (для этого

нужно раскрывать соответствующие значки «+» в поле с детальной информацией о выбранном пакете).

7. Написать в отчёте ответы на вопросы задания (для этого может потребоваться самостоятельно изучить назначение соответствующей заданию сетевой утилиты, использованной для создания трафика).
8. Поместить в отчёт скриншоты окна Wireshark, иллюстрирующие ответы из вышеуказанных п.6 и п.7.

В качестве адреса сайта в заданиях следует использовать один из следующих URL (следует выбрать один из пунктов в порядке перечисления):

- Адрес, выбранный по явному указанию преподавателя. Если преподаватель не давал соответствующих указаний, нужно использовать следующие пункты.
- Адрес сайта с домашней страницей студента. Автор страницы должен легко идентифицироваться с этой страницей по содержимому сайта.
- Адрес сайта, в название которого лексически входит фамилия студента (например: www.sidorovivan.ru).
- Адрес сайта, в котором по очереди встречаются инициалы (ФИО) студента в латинской транскрипции (например, для имени Иванов Фёдор Михайлович подойдёт адрес сайта http://ifmo.ru).

Примечание 1. При выполнении анализа HTTP-трафика не принимать во внимание HTTP-запрос и HTTP-ответ для файла *favicon.ico*. Появление ссылки на данный файл означает, что браузер автоматически запрашивает сервер о наличии значка веб-сайта, который отображается браузером в адресной строке перед адресом страницы (и в некоторых других местах).

Примечание 2. Все используемые в УИР утилиты доступны как в ОС MS Windows, так и Linux, однако в примерах к заданию указывается синтаксис и ключи командной строки для MS Windows. В Linux команды будут иметь несколько иной синтаксис.

4 Порядок выполнения работы

4.1 Анализ трафика утилиты ping

Необходимо отследить и проанализировать трафик, создаваемый утилитой ping, запустив её следующим образом из командной строки:

“ping -l размер_пакета адрес_сайта_по_варианту”.

Например, “ping -l 2000 wireshark.org” (без кавычек).

В качестве “размера_пакета” необходимо поочерёдно использовать различные значения от 100 до 10000, самостоятельно выбрав шаг изменения.

По результатам анализа собранной трассы, необходимо ответить на следующие вопросы и выполнить указанные задания.

1. Имеет ли место фрагментация исходного пакета, какое поле на это указывает?
2. Какая информация указывает, является ли фрагмент пакета последним или промежуточным?
3. Чему равно количество фрагментов при передаче ping-пакетов?
4. Построить график, в котором на оси абсцисс находится размер_пакета, а по оси ординат – количество фрагментов, на которое был разделён каждый ping-пакет.
5. Как изменить поле TTL с помощью утилиты ping?
6. Что содержится в поле данных ping-пакета?

4.2 Анализ трафика утилиты tracert (traceroute)

Необходимо отследить и проанализировать трафик, создаваемый утилитой tracert (или traceroute в Linux), запустив её следующим образом из командной строки:

“tracert -d адрес_сайта_по_варианту”

Например, tracert wireshark.org.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных?
2. Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracert? Для ответа на этот вопрос нужно проследить изменение TTL при передаче по маршруту, состоящему из более чем двух хопов.
3. Чем отличаются ICMP-пакеты, генерируемые утилитой tracert, от ICMP-пакетов, генерируемых утилитой ping (см. предыдущее задание).
4. Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов?
5. Что изменится в работе tracert, если убрать ключ “-d”? Какой дополнительный трафик при этом будет генерироваться?

4.3 Анализ HTTP-трафика

Необходимо отследить и проанализировать HTTP-трафик, создаваемый браузером при посещении Интернет-сайта, заданного по варианту. В списке

захваченных пакетов необходимо проанализировать следующую пару HTTP-сообщений (запрос-ответ):

- GET-сообщение от клиента (браузера);
- ответ сервера.

Для этого в поле с детальной информацией о пакете нужно развернуть строку “HTTP”. Затем необходимо обновить страницу в браузере так, чтобы вместо «HTTP GET» был сгенерирован «HTTP CONDITIONAL GET» (так называемый «условный GET»). Условные запросы GET содержат поля If-Modified-Since, If-Match, If-Range и подобные, которые позволяют при повторном запросе не передавать редко изменяемые данные. В ответ на условный GET тело запрашиваемого ресурса передается только в том случае, если этот ресурс изменялся после даты «If-Modified-Since». Если ресурс не изменялся, сервер вернет код статуса «304 Not Modified».

По результатам анализа собранной трассы покажите, каким образом протокол HTTP передавал содержимое страницы при первичном посещении страницы и при вторичном запросе-обновлении от браузера (т.е. при различных видах GET-запросов).

4.4 Анализ DNS-трафика

Необходимо отследить и проанализировать трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

- настроить Wireshark-фильтр: “ip.addr == ваш_IP_адрес”;
- очистить кэш DNS с помощью команды ipconfig в командной строке: ipconfig /flushdns
- очистить кэш браузера;
- зайти на Интернет-сайт, заданный по варианту.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Почему адрес, на который отправлен DNS-запрос, не совпадает с адресом посещаемого сайта?
2. Какие бывают типы DNS-запросов?
3. В какой ситуации нужно выполнять независимые DNS-запросы для получения содержащихся на сайте изображений?

4.5 Анализ ARP-трафика

Необходимо отследить и проанализировать трафик протокола ARP, сгенерированный в результате выполнения следующих действий:

- очистить ARP-таблицу командой “netsh interface ip delete arpcache”

(проверить очистилась ли таблица можно с помощью команды команды “arp -a”, выводящей таблицу на экран);

- очистить кэш браузера;
- зайти на Интернет-сайт, заданный по варианту.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что означают эти адреса? Какие устройства они идентифицируют?
2. Какие MAC-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Что означают эти адреса? Какие устройства они идентифицируют?
3. Для чего ARP-запрос содержит IP-адрес источника?

4.6 Анализ трафика утилиты nslookup

Это задание является необязательным, его необходимо выполнить только для желающих получить оценку «**хорошо**» или «**отлично**». Необходимо отследить и проанализировать трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

1. Настроить Wireshark-фильтр: “ip.addr == ваш_IP_адрес”.
2. Запустить в командной строке команду “nslookup адрес_сайта_по_варианту”.
3. Дождаться отправки трёх DNS-запросов и трёх DNS-ответов (в работе нужно использовать только последние из них, т.к. первые два набора запросов/ответов специфичны для nslookup и не генерируются другими сетевыми приложениями).
4. Повторить предыдущие два шага, используя команду: “nslookup -type=NS имя_сайта_по_варианту”.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Чем различается трасса трафика в п.2 и п.4, указанных выше?
2. Что содержится в поле «Answers» DNS-ответа?
3. Каковы имена серверов, возвращающих авторитативный (authoritative) отклик?

4.7 Анализ FTP-трафика

Это задание является необязательным, его необходимо выполнить только для желающих получить оценку «**хорошо**» или «**отлично**».

Необходимо отследить и проанализировать трафик протокола FTP, сгенерированный в результате выполнения следующих действий:

- настроить Wireshark-фильтр «ftp || ftp-data»;
- скачать в браузере небольшой файл с соответствующего варианту FTP-сервера в Интернете.

В адресной строке путь к скачиваемому файлу должен начинаться с «ftp://». Адрес сайта нужно выбрать, руководствуясь правилами, указанными в пункте 3.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Сколько байт данных содержится в пакете FTP-DATA?
2. Как выбирается порт транспортного уровня, который используется для передачи FTP-пакетов?
3. Чем отличаются пакеты FTP от FTP-DATA?

4.8 Анализ DHCP-трафика

Это задание является необязательным, его необходимо выполнить для желающих получить оценку «**отлично**». Необходимо отследить и проанализировать трафик протокола DHCP, сгенерированный в результате выполнения следующих действий:

1. Убедиться, что для назначения IP-адреса на компьютере был использован DHCP и что компьютеру был назначен IP-адрес.
2. Настроить Wireshark-фильтр «bootp» (во время защиты УИР следует объяснить, почему именно такой фильтр используется для анализа DHCP-трафика).
3. Сбросить текущий IP-адрес, выданный накануне перед этим DHCP-сервером, с помощью команды:
   “ipconfig /release“.
4. Запросить новый IP-адрес с помощью команды:
   “ipconfig /renew“.
5. Повторить п.3 и п.4.

Нарисуйте временную диаграмму, иллюстрирующую последовательность обмена первыми четырьмя DHCP-пакетами Discover/Offer/Request/ACK. Укажите для каждого пакета номера портов источника и назначения. По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Чем различаются пакеты «DHCP Discover» и «DHCP Request»?
2. Как и почему менялись MAC- и IP-адреса источника и назначения в переданных DHCP-пакетах.

3. Каков IP-адрес DHCP-сервера?
4. Что произойдёт, если очистить использованный фильтр “bootp”?

4.9 Анализ Skype-трафика

Это задание является необязательным, его необходимо выполнить для желающих получить оценку «**отлично**». Необходимо отследить и проанализировать трафик Skype (или любой другой аналогичной по функциональности программы), сгенерированный в результате выполнения следующих действий:

- отправить текстовое сообщение и получить ответ;
- осуществить короткий сеанс аудио-общения;
- осуществить короткий сеанс видео-общения.

Для упрощения анализа передачи различных видов трафика Скайпом (тест, аудио, видео) можно независимо собрать трассы трафика для каждого из трёх перечисленных пунктов, останавливая и возобновляя захват трафика так, чтобы получить три отдельных файла. По результатам анализа трёх собранных видов трасс трафика ответьте на следующие вопросы.

1. Чем различаются пакета разных видов Skype-трафика (текст, аудио, видео)?
2. Какой Wireshark-фильтр следует использовать для независимой идентификации Skype-трафика разных видов (текст, аудио, видео)?

Примечание. При выполнении п. 2.4.9 вместо Skype можно использовать любое другое аналогичное по функциональности программное обеспечение (Yahoo Messenger, MSN, Tox, «Mail.ru Агент» и любые другие)

5 Требования к содержанию отчёта

В работе требуется проанализировать сетевой трафик, захваченный с помощью программы Wireshark. В отчёте, предоставляемом в электронном или бумажном виде, следует привести скриншоты, иллюстрирующие ответы на поставленные в задании вопросы. Каждый скриншот должен иметь поясняющий текст, подробно раскрывающий содержание ответа на соответствующие вопросы.

Также в отчёте необходимо привести структуру наблюдаемых пакетов (как запросов, так и ответов), коротко описав назначение всех заголовков всех уровней с учётом порядка инкапсуляции. Стоит привести описание только тех пакетов, которые существенно различаются структурно (однотипные похожие пакеты приводить не надо), либо имеют непосредственное отношение к ответам на вопросы задания.

При защите отчёта необходимо иметь при себе сохраненную версию захваченного трафика на flash-носителе в формате *pcap* (так называемую трассу, или дамп, трафика).

6 Контрольные вопросы для самопроверки

При подготовке к защите задания следует руководствоваться следующим примерным перечнем вопросов и задач для самостоятельной проработки.

1. Что такое OSI-модель и для чего она нужна?
2. Перечислите уровни OSI-модели и дайте им краткую характеристику.
3. Какие преимущества даёт многоуровневая архитектура OSI-модели? Какие бы возникли сложности, если процесс передачи данных по сети был одноуровневым?
4. Если перед отправкой данных выполняется их сжатие, то на каком уровне OSI-модели следует выполнять эту операцию?
5. Как соотносится модель OSI с реальной структурой существующих стеков протоколов?
6. Функции каких уровней OSI-модели выполняет протокол TCP?
7. Что схожи и чем различаются протоколы UDP, TCP, SCTP и DCCP?
8. Приведите примеры протоколов (или целых стеков), которые нарушают канонические требования OSI-модели.
9. Покажите на примере Wireshark-скриншотов, какие поля в заголовках разных уровней и как именно соотносятся с функциями соответствующих уровней OSI-модели.
10. Каковы основные функциональные возможности программы Wireshark?
11. Какие существуют аналоги программы Wireshark?
12. Каким образом можно перенести (экспортировать) данные о пакетах в Wireshark-трассе в таблицу MS Excel?
13. Какие уровни OSI-модели “умеет“ анализировать программа Wireshark? Приведите конкретные примеры протоколов.
14. Используя шестнадцатеричное представление пакетов в окне Wireshark укажите в какой последовательности передаются байты в сеть на примере IP-адреса в заголовке пакета. Для ответа на этот вопрос см. самостоятельно темы https://ru.wikipedia.org/wiki/Порядок_байтов или https://en.wikipedia.org/wiki/Endianness.
15. Какими средствами можно отправить в сеть пакеты, записанные в файле-трассе pcap?

16. Сколько различных MAC- и IP-адресов можно закодировать, используя отведённое для них количество байт в заголовках?
17. Какой процент избыточности вносят заголовки разных уровней в передаваемые сообщения?
18. Какие поля в заголовках разных уровней вне поля данных можно использовать в целях стеганографии, т.е. для передачи скрытых (дополнительных) данных так, чтобы добавленные данные не мешали передаче пакетов. Какие условия должны при этом соблюдаться?
19. Что означает цветовая дифференциация пакетов в Wireshark?