Информационные сети и телекоммуникации — 2016

Лабораторная работа № 4 «Диагностика сетевых протоколов»

Цель работы

Наблюдение практической реализации сетевых протоколов. Изучение механизма работы протоколов сетевого стека TCP/IP. Ознакомление с методами и средствами диагностики сетевых протоколов.

Задание на лабораторную работу

Пункты, в которых требуется перехватывать сетевой трафик, отмечены знаком ★.

1. **Определить сетевые настройки машины**
   - 1.1. Командой `ipconfig /all` (Windows), `ip address show` (Linux) или через GUI определить и зафиксировать в отчете адрес IP, MAC-адрес, маску подсети, адрес шлюза по умолчанию.
   - 1.2. Вычислить номер сети и зафиксировать его в отчете.

2. **Пронаблюдать работу протоколов канального уровня**
   - 2.1. Просмотреть локальный кэш ARP командой `arp -a` (Windows) или `sudo arp -n` (Linux).
   - 2.2. Программой, написанной в ходе ЛР № 1, отправить несколько сообщений на любой адрес в локальной сети, кроме шлюза по умолчанию. (Принимать эти сообщения не нужно, но отправлять следует с разрешенных портов.) ★
   - 2.3. Проанализировать все перехваченные пакеты ARP (фильтр `arp`). Отыскать в заголовке кадра Ethernet и в пакете ARP все основные поля, рассмотренные в лекциях. Объяснить их наблюдаемые значения. Понять назначение каждого пакета.
   - 2.4. Повторить пункт 2.1 и объяснить изменения.

3. **Изучить функционирование сетевого уровня**
   - 3.1. Напечатать таблицу маршрутизации командой `route print` (Windows) или `ip route show table all` (Linux) . Объяснить значение каждой записи. Пример: «обеспечивает отправку пакетов широковещательной рассылки в сеть X с интерфейса Y».
   - 3.2. Определить маршрут до узла 8.8.8.8 командой `tracert 8.8.8.8` (Windows) или `sudo traceroute -I 8.8.8.8` (Linux) ★
   - 3.3. Проанализировать перехваченные пакеты ICMP (фильтр `icmp`), а также содержащие их пакеты IP, и сделать вывод о методе работы команды.

3.4. Пронаблюдать разрешение символьного имени сервера в адрес IP:

3.4.1. Очистить системный кэш DNS командой `ipconfig /flushdns` (Windows) или `sudo systemctl restart networking` (Debian 8).

3.4.2. Определить адрес IP сервера кафедры (`uii.mpei.ru`) командой `nslookup uii.mpei.ru` (в любой ОС) ★

3.4.3. Проанализировать перехваченные пакеты DNS (фильтр `dns`): определить назначение каждого пакета и сопоставить данные в них с выводом команды (указать, из каких пакетов и какие получены сведения).

4. Пронаблюдать работу протокола TCP транспортного и сеансового уровня

4.1. Воспользовавшись программой из ЛР № 2 или 3, выполнить действия: ★

1) запустить сервер;
2) подключиться к серверу, загрузить файл размером в десятки байт;
3) загрузить файл размером порядка сотен килобайт (например, исполняемый файл сервера), отключиться;
4) не останавливая сервер, повторить пункт 2) и отключиться;
5) остановить сервер.

Примечание. Сервер и клиент следует запускать на разных машинах, договорившись с соседней бригадой.

4.2. Проанализировать записанные сеансы TCP (фильтр по полю `tcp.port`):

4.2.1. При помощи инструмента Colorize Conversation (в контекстном меню пакета), подсветить сеансы связи с каждым клиентом.

4.2.2. В каждом сеансе отыскать пакеты запросов и ответов.

4.2.3. Выделить характерные пакеты в начале и в конце каждого сеанса (трехфазное рукопожатие и корректное завершение сеанса).

4.2.4. Отыскать участок сеанса, в котором проходило согласование размера скользящего окна. Выяснить, принадлежала ли инициатива приемнику или отправителю.

5. Пронаблюдать работу HTTP

5.1. Запустить в дополнение к Wireshark web-браузер и перейти на страницу дисциплины. Включить панель разработчика (*Shift+F5* в Firefox и Chrome, *F12* в Internet Explorer) и выбрать на ней вкладку Network.

5.2. Обновить страницу (*F5* или кнопка *Reload* на панели разработчика). Пронаблюдать совершаемые web-браузером запросы, включая адреса ресурсов, заголовки запросов и ответов на них. Идентифицировать элементы web-страницы, получаемые по запросам. ★

5.3. Просмотреть запросы и ответы HTTP в Wireshark в текстовом виде (packet bytes pane внизу окна) и сличить текст ответа с отображаемым на панели разработчика (вкладка *Network,* раздел *Response* при выборе запроса).

5.4. Перейти по адресу https://example.com/. ★

5.5. Пронаблюдать установление сеанса TLS (HTTPS) в Wireshark и установить, на каком уровне протоколов шифруются данные и начиная с какого этапа.

Указания к выполнению лабораторной работы

Перед выполнением работы нужно ознакомиться с кратким руководством по Wireshark.

При повторном захвате трафика в пунктах 2 и 5.2 следует повторно очищать кэш ARP, DNS и web-браузера соответственно. Записи из кэша ARP удаляются в WIndows и Linux командой `arp -d` *`адрес-IP`* от имени администратора.

Панель разработчика в web-браузерах

Панель разработчика (Developer Tools) служит для анализа и отладки web-приложений со стороны клиента. Базовый набор функций панели разработчика близок во всех обозревателях. Возможен анализ и изменение свойств любого элемента страницы, отладка сценариев JavaScript, наблюдение сеансов HTTP при загрузке документа и его фрагментов (изображений и т. п.). Простого способа сохранить результаты как текст нет.

Элементарные клиенты сетевых протоколов

При отсутствии собственных программ почти для любого протокола возможно отыскать готовые клиенты с элементарной функциональностью. В данной ЛР мог бы пригодиться PacketSender (клиент и сервер TCP и UDP), в ОС *nix те же задачи решает `netcat`.

Контрольные вопросы

1. Что такое маска подсети, для чего и как она используется? Приведите пример.
2. Как маск*и*рованием выделить в сети 192.0.2.0/24 на 7 сетей, каждая из которых могла бы вместить не менее 17-и узлов?
3. Как может быть отпределен MAC-адрес при известном адресе IP, и наоборот? В каких случаях эти процедуры применяются и какие протоколы используются?
4. Как по сообщению ICMP Time Exceeded получатель определяет, какой именно пакет не удалось доставить?
5. Узлы не обязаны отвечать на сообщения ICMP Echo Request. Предложите способ работы программы `tracert` (`traceroute`) для этого случая.
6. Может ли адрес отправителя в заголовке пакета IP не принадлежать узлу, передавшему пакет? Если да, в каких случаях это делается, если нет, то почему?
7. Что такое скользящее окно TCP (sliding window)? На что влияет его размер, из каких соображений и когда он устанавливается?
8. Какими последовательностями пакетов начинается и заканчивается сеанс TCP, какие задачи при этом решаются?
9. Какие виды web-приложений существуют (с точки зрения обслуживания запросов HTTP) и какой из них применялся в п. 5 данной ЛР?
10. Из каких компонент состоит универсальный идентификатор ресурса (URI)? Как сервер HTTP получает каждую из них и для чего они используются?