Курс «Информационные сети и телекоммуникации»

Лабораторная работа № 5 «Диагностика сетевых протоколов»

Цель работы

Наблюдение практической реализации сетевых протоколов. Изучение механизма работы протоколов сетевого стека TCP/IP и структуры их пакетов. 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` 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 командой `arp -a` локальный кэш ARP.
   - 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 командой `route print` таблицу маршрутизации. Объяснить значение каждой записи, её задачу. 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` ★
   - 3.3. Проанализировать перехваченные пакеты ICMP (фильтр `icmp`), а также содержащие их пакеты IP, и сделать вывод о методе работы `tracert`.
   - 3.4. Запустить три экземпляра программы из ЛР № 4, задва в каждом из них один и тот же порт. Включить их в одну группу многоадресной рассылки,

отправить из любого экземпляра сообщение в группу, затем штатно завершить работу всех экземпляров (командой `/quit`, см. ЛР № 4). ★

3.5. Проанализировать перехваченные пакеты IGMP (фильтр `igmp`) и UDP (фильтры по полю `udp.srcport`), а также содержащие их пакеты IP. Сопоставить количества входящих и исходящих пакетов UDP. Объяснить происхождение каждого пакета UDP и значение каждого пакета IGMP.

3.6. Пронаблюдать разрешение символьного имени сервера в адрес IP:

3.6.1. Очистить системный кэш DNS командой `ipconfig /flushdns`.

3.6.2. Определить адрес IP сервера кафедры (`uii.mpei.ru`) командой `nslookup uii.mpei.ru` ★

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

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.

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 курса. Включить панель разработчика (*Shift+F5* в Firefox и Chrome, *F12* в Internet Explorer) и выбрать на ней вкладку Network.

5.2. Перейти на страницу курса. 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. Воспользовавшись инструментом Colorize Conversation (в контекстном меню пакета), подсветить сеанс TCP, содержащий запрос и ответ HTTP.

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

5.5. Пронаблюдать установление сеанса TLS (HTTPS) в Wireshark и описать отличия от незащищенного соединения HTTP.

Указания к выполнению лабораторной работы

Любая сетевая активность во время выполнения должна быть сведена к минимуму. Например, стоит отключить web-браузер (кроме п. 5), клиенты «облачных» хранилищ и сетевых дисков, программы для обмена файлами и прочее ПО, использующее сеть.

При повторном захвате трафика в пунктах 2, 3.6 и 5.2 следует повторно очищать кэш ARP, DNS и web-браузера соответственно. Записи из кэша ARP в ОС Windows удаляются командой `arp -d` *`адрес-IP`* от имени администратора (см. ниже).

Анализатор сетевого трафика Wireshark

Программа Wireshark предназначена для захвата, записи и анализа сетевого трафика. Перехваченные пакеты известных протоколов представляются как в виде набора байт, так и в разобранном виде; могут быть отобраны с помощью фильтров (по значению поля, по размеру, по принадлежности к сеансу TCP и другим).

Запись трафика может быть сохранена в файл `*.pcap`, который позднее возможно открыть в Wireshark или иной программе для просмотра и анализа. Создавать такие файлы способна не только Wireshark, но и другие программы, например, `tcpdump`.

Перехват пакетов запускается нажатием кнопки «Start». При наличии нескольких сетевых интерфейсов (подключений в Windows) можно выбрать нужные из списка.

Описание интерфейса главного окна поясняет назначение основных его областей, а также содержит ссылки на подробные описания каждой из них, включая правила составления выражений-фильтров.

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 нет.

Команды в ОС семейства *nix

Определение маршрута до узла возможно командой `traceroute -I` *`адрес`*. Ключ `-I` нужен затем, чтобы в качестве пробных пакетов использовались ICMP Echo Request, а не дейтаграммы UDP.

Сброс кэша DNS в большинстве дистрибутивов Linux выполняется перезапуском службы `nscd` командой `systemctl restart nscd` или `service nscd restart`, в OS X — командой `dscacheutil -flushcache`. См. также развернутую заметку.

И `traceroute`, и перезапуск службы требуется выполнять с правами администратора.

Работа вне лаборатории

Ядро Wireshark, библиотека *libpcap,* в ОС Windows не способна перехватывать пакеты, передаваемые между интерфейсами одной машины. Простейшее решение — вынудить ОС прокладывать маршрут для таких пакетов через другую машину; подходит шлюз по умолчанию. В ОС семейства *nix проблема отсутствует.

Внешний адрес машины *(свой-IP)* и адрес шлюза можно узнать командой `ipconfig` или в диалоге свойств сетевого подключения.

Проложить маршрут, действующий до перезагрузки, можно следующей командой:

```
route add  свой-IP mask 255.255.255.255  IP-шлюза  metric 1
```

Изменение таблицы маршрутизации разрешено только администратору. В Windows 7 и более поздних версиях запуск командной строки администратора доступен из GUI.

По окончании экспериментов следует удалить этот неоптимальный, вредный варишрут:

```
route delete  свой-IP  IP-шлюза
```

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 применяются `netcat`, `iperf` и др.

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?
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?