ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ им. проф. М. А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

С. С. Владимиров

ПРОТОКОЛЫ, СЕРВИСЫ И УСЛУГИ В IP-СЕТЯХ

Лабораторный практикум

СПб ГУТ)))

Санкт-Петербург
2019

УДК xxx.xxx.x (xxx)
ББК 00.00 x00
В 57

Рецензент
профессор кафедры СС и ПД,
доктор технических наук *О. С. Когновицкий*

Рекомендован к печати редакционно-издательским советом СПбГУТ

Владимиров, С. С.

В 57 Протоколы, сервисы и услуги в IP-сетях : лабораторный практикум / С. С. Владимиров ; СПбГУТ. — СПб, 2019. — 16 с.

Учебное пособие призвано ознакомить студентов старших курсов с интернет-технологиями. Представленный материал служит справочным и методическим пособием при выполнении курса практических лабораторных работ по дисциплине «Протоколы, сервисы и услуги в IP-сетях».

Предназначено для студентов, обучающихся по направлениям 11.03.02 «Инфокоммуникационные технологии и системы связи», 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия».

УДК xxx.xxx.x (xxx)
ББК 00.00 x00

Содержание

Лабораторная работа 1
Изучение протоколов электронной почты

1.1. Цель работы

Лабораторная работа ставит цели закрепления теоретического материала по протоколам и программному обеспечению служб электронной почты. В рамках заданий данной лабораторной работы рассматриваются протоколы SMTP, POP3, IMAP4 и принципы подключения к соответствующим серверам.

1.2. Задачи

Ознакомившись с принципами работы основных протоколов электронной почты, с помощью клиента службы `telnet` отправить электронное письмо на удаленный почтовый сервер и получить письмо с почтового сервера по протоколам POP3 и IMAP4.

1.3. Порядок выполнения лабораторной работы

IP-адрес сервера: `172.16.100.88`

Адрес e-mail: `student_01@opdsnet`
(по номеру ПК) `student_02@opdsnet`
`...`
`student_16@opdsnet`

Пароль: `student`

Для установления соединения и передачи/приема письма использовать утилиту `telnet`.

1.3.1. Отправка письма по протоколу SMTP

1. Начать захват пакетов при помощи анализатора протоколов (`wireshark`, `tcpdump`). Захват проводить по фильтру (IP-адреса источника и получателя, протокол TCP, номер порта со стороны сервера; для `tcpdump` дополнительно указать размер пакета 1500 байт и запись в двоичный файл).
2. Соединиться с SMTP-сервером кафедры.
3. Сформировать и передать письмо.
4. Разорвать соединение.
5. Сохранить результаты работы в текстовый файл.
6. По захваченным пакетам построить диаграмму Flow Graph с помощью `wireshark`. Диаграмму сохранить либо в виде текстового файла либо в виде изображения.

1.3.2. Получение письма по протоколу POP3

1. Начать захват пакетов при помощи анализатора протоколов (`wireshark`, `tcpdump`). Захват проводить по фильтру (IP-адреса источника и получателя, протокол TCP, номер порта со стороны сервера; для `tcpdump` дополнительно указать размер пакета 1500 байт и запись в двоичный файл).
2. Соединиться с POP3-сервером кафедры.
3. Вывести на экран список писем.
4. Получить полный текст переданного почтового сообщения.
5. Удалить переданное почтовое сообщение.
6. Разорвать соединение.
7. Сохранить результаты работы в текстовый файл.
8. По захваченным пакетам построить диаграмму Flow Graph с помощью `wireshark`. Диаграмму сохранить либо в виде текстового файла либо в виде изображения.

1.4. Содержание отчета

1. Титульный лист согласно приложению.
2. Цель работы.
3. Результаты работы с протоколом SMTP.
4. Диаграмма Flow Graph для SMTP.
5. Результаты работы с протоколом POP3.
6. Диаграмма Flow Graph для POP3.

1.5. Контрольные вопросы

1. Структура электронного сообщения. Назначение полей заголовка электронного письма, полученного по протоколу POP3.
2. Адресация и маршрутизация электронной почты в Internet.
3. Структура электронной почты в Internet.
4. Протокол POP3.
5. Протокол SMTP.
6. Протокол LMTP.

Лабораторная работа 2
Ознакомление с системой и протоколом DNS

2.1. Цель работы

Лабораторная работа ставит цели закрепления теоретического материала по протоколам и программному обеспечению системы доменных имен.

2.2. Порядок выполнения лабораторной работы

2.2.1. Разрешение адресов в системе DNS с использованием различных утилит системы Linux

1. Запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика DNS.

2. В терминале последовательно выполнить разрешение доменных имен `yandex.ru` и `www.yandex.ru` с использованием трех утилит:

   а) `host`

   б) `nslookup`

   в) `dig`

3. Остановить захват пакетов в анализаторе протоколов Wireshark.

4. Проанализировать вывод команд и перехваченные пакеты.

5. В отчете привести последовательно вывод каждой команды, сопроводив его соответствующими перехваченными пакетами и выводами по работе программ.

2.2.2. Получение ресурсных записей различных типов с использованием утилиты `host`

1. Запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика DNS.

2. Последовательно получить от сервера DNS следующие ресурсные записи для доменного имени `yandex.ru`:

   а) адреса IPv4

   б) адреса IPv6

   в) почтовые серверы

   г) серверы DNS

   д) авторитетный сервер для доменного имени

Для получения справки о работе с программой `host` необходимо использовать команду

```
user@host:[~]$ man host
```

3. Остановить захват пакетов в анализаторе протоколов Wireshark.

4. Проанализировать вывод команд и перехваченные пакеты.

5. В отчете привести последовательно вывод каждой команды, сопроводив его соответствующими перехваченными пакетами и выводами по работе программ.

2.2.3. Проведение обратного запроса DNS с использованием утилиты `host`

1. Запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика DNS.

2. Провести обратный запрос DNS для IPv4-адреса `77.88.55.80`

3. Провести обратный запрос DNS для IPv6-адреса `2a02:6b8:a::a`. При формировании команды запроса руководствоваться примером ресурсной записи:

```
IPv6-address: 2001:0db8::1:2345
DNS-record:
   5.4.3.2.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d
   .0.1.0.0.2.ip6.arpa. IN PTR host1.example.net.
```

4. Остановить захват пакетов в анализаторе протоколов Wireshark.

5. Проанализировать вывод команд и перехваченные пакеты.

6. В отчете привести последовательно вывод каждой команды, сопроводив его соответствующими перехваченными пакетами и выводами по работе программ.

2.2.4. Получение всех ресурсных записей для определенного доменного имени с использованием утилиты `host`

1. Запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика: IP-адреса ПК и сервера DNS, протоколы (TCP или UDP).

2. Выполнить запрос «ресурсной записи» `ANY` для доменного имени `yandex.ru`.

3. Остановить захват пакетов в анализаторе протоколов Wireshark.

4. Проанализировать вывод команд и перехваченные пакеты.

5. В отчете привести вывод команды, сопроводив его соответствующими перехваченными пакетами и выводами по процедуре получения записи.

2.3. Содержание отчета

1. Заголовок согласно приложению.

2. Цель работы.

3. Результаты работы по каждому из пунктов работы с соответствующими выводами.

2.4. Контрольные вопросы

1. Что такое система доменных имён?

2. Для чего используется файл hosts?
3. Каковы ключевые характеристики DNS?
4. Что такое домен и поддомен?
5. Что такое корневой домен?
6. Что такое корневые серверы DNS?
7. Что такое псевдоинтернациональные домены?
8. Что такое рекурсия в DNS?
9. Как выполняется DNS-запрос?
10. Что такое обратный DNS-запрос?

Лабораторная работа 3
Моделирование сетевых DoS-атак

3.1. Цель работы

Целью данной работы является ознакомление с методиками проведения некоторых сетевых атак типа «отказ в обслуживании», а также краткое ознакомление с программой формирования сетевых пакетов Scapy.

3.2. Порядок выполнения лабораторной работы

3.2.1. Моделирование сетевой DoS-атаки типа SYN-Flood и сетевой DoS-атаки с использованием отраженных TCP-пакетов

1. На рабочем ПК запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика: TCP, IP-адрес сервера.
2. На атакуемом ПК запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика: TCP, IP-адрес сервера.
3. В терминале на рабочем ПК запустить утилиту Scapy командой

```
user@host:[~]$ sudo scapy
```

4. В командной строке Scapy сформировать и передать на сервер TCP-пакет, инициирующий установление TCP-соединения, так, чтобы ответ был направлен на атакуемый ПК. Использовать шаблон

```
>>> send(IP(src="192.168.1.14",dst="192.168.1.1")/TCP(sport
   =45010,dport=80,flags="S"))
```

Возможные параметры протоколов IPv4 и TCP можно увидеть, использовав следующие команды:

```
>>> ls(IP)
>>> ls(TCP)
```

5. Остановить захват пакетов в анализаторе протоколов Wireshark на рабочем и атакуемом ПК.
6. Проанализировать вывод команд и перехваченные пакеты.
7. В отчете привести последовательно вывод каждой команды, сопроводив его соответствующими перехваченными пакетами и выводами по проведенной атаке.

3.2.2. Моделирование сетевой DoS-атаки с использованием отраженных запросов DNS

1. На рабочем ПК запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика: UDP, IP-адрес DNS-сервера.
2. На атакуемом ПК запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика: UDP, IP-адрес DNS-сервера.

3. В терминале на рабочем ПК запустить утилиту Scapy командой

```
user@host:[~]$ sudo scapy
```

4. В командной строке Scapy сформировать и передать на DNS-сервер запрос ресурсной записи ANY для домена google.ru, так, чтобы ответ был направлен на атакуемый ПК. Использовать шаблон

```
>>> send(IP(src="192.168.1.14",dst="192.168.1.1")/UDP(dport
    =53,sport=45100)/DNS(id=1393,rd=1,qd=DNSQR(qname="ya.ru",
    qtype=255)))
```

Возможные параметры протоколов UDP и DNS можно увидеть, использовав следующие команды:

```
>>> ls(UDP)
>>> ls(DNS)
```

5. Повторить атаку, задав диапазон портов отправителя:

```
sport=(45100,45150)
```

6. Остановить захват пакетов в анализаторе протоколов Wireshark на рабочем и атакуемом ПК.

7. Проанализировать вывод команд и перехваченные пакеты. Посчитать коэффициент усиления атаки, приняв во внимание долю ICMP-ответов об ошибке, отправляемых атакованным хостом.

8. В отчете привести последовательно вывод каждой команды, сопроводив его соответствующими перехваченными пакетами и выводами по проведенной атаке.

3.3. Содержание отчета

1. Заголовок согласно приложению.
2. Цель работы.
3. Результаты работы по каждому из пунктов работы с соответствующими выводами.

3.4. Контрольные вопросы

1. Что такое сетевая DoS-атака?
2. Принцип атаки SYN-Flood.
3. Принцип атаки посредством отраженных запросов DNS.

Лабораторная работа 4
Ознакомление с сетевой утилитой `netcat` и туннелированием TCP соединений

4.1. Цель работы

Целью лабораторной работы является ознакомление с работой сетевой утилиты `netcat`, закрепление теоретического материала по работе протоколов TCP и SSH, а также получение навыков в организации безопасного туннелирования TCP-соединений.

4.2. Порядок выполнения лабораторной работы

Схема работы показана на рис. 4.1.

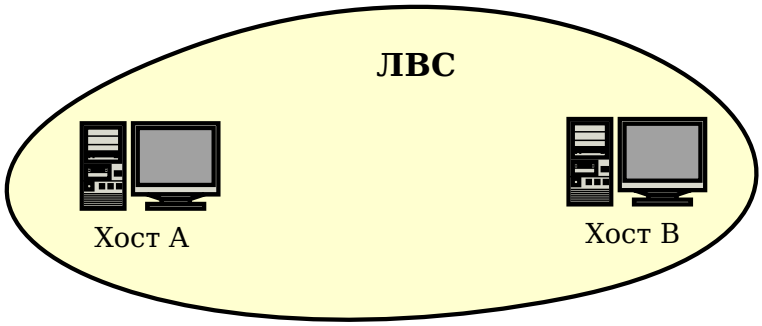

Рис. 4.1. Схема выполнения лабораторной работы

4.2.1. Установление простого соединения между двумя ПК с использованием утилиты `netcat`

1. Запустить на ПК **A** анализатор протоколов Wireshark и указать фильтр для перехвата трафика: TCP, IP-адреса рабочих ПК.
2. В терминале ПК **A** запустить утилиту `netcat` в пассивном режиме (флаг `-l`), указав для прослушивания номер порта более 20000. Использовать шаблон:

```
user@host:[~]$ nc -l -p port-number
```

3. В терминале ПК **B** запустить утилиту `netcat`, установив соединение с ПК **A**. Использовать шаблон:

```
user@host:[~]$ nc IP-address port-number
```

4. После установления соединения, текст, набранный в терминале, по нажатию клавиши Enter будет пересылаться на другой ПК. Убедиться в этом, передав по установленному соединению несколько произвольных осмысленных фраз в разные стороны. После этого завершить соединение, нажав комбинацию Ctrl + C .
5. Остановить захват пакетов в анализаторе протоколов Wireshark.

6. Проанализировать вывод команд и перехваченные пакеты.

7. В отчете привести команды установления соединения, сопроводив их соответствующими перехваченными пакетами и выводами по работе программы.

4.2.2. Передача файла между двумя ПК с использованием утилиты `netcat`

1. Запустить на ПК **A** анализатор протоколов Wireshark и указать фильтр для перехвата трафика: TCP, IP-адреса рабочих ПК.

2. Создать на ПК **A** текстовый файл, содержащий номер группы и ФИО, выполняющих работу учащихся.

3. В терминале ПК **A** запустить утилиту `netcat` в пассивном режиме (флаг `-l`), указав для прослушивания номер порта более 20000 и передав в нее созданный файл при помощи утилиты `cat` и механизма конвейера. Использовать шаблон:

```
user@host:[~]$ cat filename | nc -l -p port-number
```

4. В терминале ПК **B** запустить утилиту `netcat`, установив соединение с ПК **A**, и перенаправить ее вывод в текстовый файл. Использовать шаблон:

```
user@host:[~]$ nc IP-address port-number > filename
```

5. Завершить соединение, нажав комбинацию Ctrl + C . Убедиться в том, что файл передан полностью.

6. Остановить захват пакетов в анализаторе протоколов Wireshark.

7. Проанализировать вывод команд и перехваченные пакеты.

8. В отчете привести команды установления соединения, сопроводив их соответствующими перехваченными пакетами и выводами по работе программы.

4.2.3. Выполнение команд на удаленном ПК с использованием утилиты `netcat`

1. Запустить на ПК **A** анализатор протоколов Wireshark и указать фильтр для перехвата трафика: TCP, IP-адреса рабочих ПК.

2. В терминале ПК **A** запустить утилиту `netcat` в пассивном режиме (флаг `-l`), указав для прослушивания номер порта более 20000 и вызвав в качестве получателя/отправителя данных командный процессор `Bash`. Использовать шаблон:

```
user@host:[~]$ nc -l -p port-number -e /bin/bash
```

3. В терминале ПК **B** запустить утилиту `netcat`, установив соединение с ПК **A**. Использовать шаблон:

```
user@host:[~]$ nc IP-address port-number
```

4. После установления соединения, команды, набранные в терминале ПК **B**, по нажатию клавиши Enter будет пересылаться на ПК **A** и передаваться в командный процессор. Убедиться в этом, последовательно передав по установленному соединению с ПК **B** на ПК **A** следующие команды:

```
uname -a
```

```
/sbin/ifconfig
```

Убедившись, что вывод команд соответствует ПК **A**, завершить соединение, нажав комбинацию Ctrl + C .

5. Остановить захват пакетов в анализаторе протоколов Wireshark.

6. Проанализировать вывод команд и перехваченные пакеты.

7. В отчете привести команды установления соединения, сопроводив их соответствующими перехваченными пакетами и выводами по работе программы.

4.2.4. Создание SSH-туннеля клиент-сервер (local port forwarding)

1. Запустить на ПК **A** анализатор протоколов Wireshark и указать фильтр для перехвата трафика: TCP, IP-адреса рабочих ПК.

2. В терминале ПК **A** запустить утилиту `netcat` в пассивном режиме (флаг `-l`), указав для прослушивания номер порта более 20000 (port-A). Этот процесс будет играть роль сервера.

3. В терминале ПК **B** запустить программу `ssh`, установив SSH-туннель к ПК **A**. Использовать шаблон:

```
user@host:[~]$ ssh -N -L port-B:IP-A:port-A username@IP-A
```

Локальный порт (port-B) должен иметь номер более 20000 и не должен быть равен номеру удаленного порта (port-A).

4. В другом терминале ПК **B** запустить утилиту `netcat`, указав ей связаться с локальным портом (port-B). Использовать шаблон:

```
user@host:[~]$ nc 127.0.0.1 port-B
```

5. После установления соединения передать несколько произвольных осмысленных фраз в разные стороны. После этого завершить соединение `netcat` и закрыть SSH-туннель, нажав в соответствующих терминалах ПК **B** комбинацию Ctrl + C .

6. Остановить захват пакетов в анализаторе протоколов Wireshark.

7. Проанализировать вывод команд и перехваченные пакеты. Сравнить с перехваченными пакетами из пункта 4.2.1.

8. В отчете привести команды установления соединения, сопроводив их соответствующими перехваченными пакетами и выводами по работе программы.

4.2.5. Создание SSH-туннеля сервер-клиент (remote port forwarding)

1. Запустить на ПК **A** анализатор протоколов Wireshark и указать фильтр для перехвата трафика: TCP, IP-адреса рабочих ПК.
2. В терминале ПК **A** запустить утилиту `netcat` в пассивном режиме (флаг `-l`), указав для прослушивания номер порта более 20000 (port-A). Этот процесс будет играть роль сервера.
3. В другом терминале ПК **A** запустить программу `ssh`, установив SSH-туннель к ПК **B**. Использовать шаблон:

```
user@host:[~]$ ssh -N -R port-B:IP-A:port-A username@IP-B
```

Удаленный порт (port-B) должен иметь номер более 20000 и не должен быть равен номеру локального порта (port-A).

4. В терминале ПК **B** запустить утилиту `netcat`, указав ей связаться с локальным портом (port-B).
5. После установления соединения передать несколько произвольных осмысленных фраз в разные стороны. После этого завершить соединение `netcat` и закрыть SSH-туннель, нажав в соответствующих терминалах ПК **A** и **B** комбинацию Ctrl + C .
6. Остановить захват пакетов в анализаторе протоколов Wireshark.
7. Проанализировать вывод команд и перехваченные пакеты. Сравнить с перехваченными пакетами из пункта 4.2.1.
8. В отчете привести команды установления соединения, сопроводив их соответствующими перехваченными пакетами и выводами по работе программы.

4.3. Содержание отчета

1. Заголовок согласно приложению.
2. Цель работы.
3. Результаты работы по каждому из пунктов работы с соответствующими выводами.

4.4. Контрольные вопросы

1. Утилита netcat.
2. SSH-туннелирование.

Приложение

Правила оформления отчета к лабораторным работам

1. Структура отчета должна соответствовать требованиям представленным в соответствующем пункте лабораторной работы.
2. Размер основного шрифта отчета: 11–12 pt.
3. Заголовок отчета должен иметь вид:

Отчет к лабораторной работе №1
Изучение принципов работы утилит
для исследования и мониторинга состояния сети

Группа: ГР-00
Студент: Пупкин В. И.

Цель работы: ...

4. Результаты работы консольных программ (листинги), сами запускаемые команды и диаграммы, отображаемые в текстовом виде, должны быть оформлены моноширинным шрифтом (Courier New, Lusida Console, FreeMono и т. п.). Они должны вмещаться в ширину страницы (шрифт можно уменьшать до 9 pt). Если ширины вертикально расположенного листа A4 не хватает, то можно разместить диаграмму на нескольких горизонтально расположенных листах A4.

Например:

```
student@comp:~\$ ping -c 4 www.ya.ru
PING ya.ru (87.250.250.3) 56(84) bytes of data.
64 bytes from www.yandex.ru (87.250.250.3): icmp_seq=1 ttl=52 time=16.8 ms
64 bytes from www.yandex.ru (87.250.250.3): icmp_seq=2 ttl=52 time=16.8 ms
64 bytes from www.yandex.ru (87.250.250.3): icmp_seq=3 ttl=52 time=18.7 ms
64 bytes from www.yandex.ru (87.250.250.3): icmp_seq=4 ttl=52 time=13.5 ms

--- ya.ru ping statistics ---
4 packets transmitted, 4 received, 0\% packet loss, time 3014ms
rtt min/avg/max/mdev = 13.542/16.484/18.759/1.874 ms
```

5. Текст на диаграммах и графиках должен быть свободно читаем.
6. На графиках должны быть подписаны оси и единицы измерения.

Владимиров Сергей Сергеевич

ПРОТОКОЛЫ, СЕРВИСЫ И УСЛУГИ В IP-СЕТЯХ

Лабораторный практикум

Редактор *Х. Х. Хxxxxxxxxx*

План изданий 20ХХ г., п. ХХ

Подписано к печати ХХ.ХХ.20ХХ
Объем Х,ХХ печ. л. Тираж ХХ экз. Заказ ХХХ

Редакционно-издательский отдел СПбГУТ
193232 СПб., пр. Большевиков, 22
Отпечатано в СПбГУТ