Содержание

Описание

Данная работа представляет собой реализацию набора программного обеспечения, который собирает и отображает статистику по трафику на заданном сетевом интерфейсе.

Набор ПО состоит из двух утилит:

1. `udp-sniff` - читает данные с сетевого интерфейса и собирает статистику (количество пакетов и байтов) по пакетам. Сбор статистики ведётся по UDP пакетам по указанным при запуске утилиты параметрам: IP-адрес источника, IP-адрес назначения, порт источника, порт назначения.
2. `print-stats` - получает собранную статитику у первой утилиты и выводит её на экран.

Варианты запуска

Первая утилита реализована в двух вариантах, различающихся в способе передачи статистики между двумя потоками (потоком считывания пакетов с интерфейса (`sniff_packets`) и потоком, передающим статистику по запросу (`provide_stats`)):

1. `sniff_packets` проверяет параметры пакетов и для подходящих **передаёт статистику во второй поток.** `provide_stats` **суммирует статистику** и отдаёт её по запросу.

   Здесь механизмом синхронизации потоков выступает канал (pipe): первый поток записывает прошедший по параметрам пакет в канал, второй - читает из него.

2. `sniff_packets` проверяет параметры пакетов и для подходящих **суммирует статистику.** `provide_stats` отдаёт её по запросу.

   В этом случае суммируемая статистика передаётся через глобальную переменную, а механизмом синхронизации выступает мьютекс. Первый поток добавляет статистику к глобальной переменной, второй - копирует в локальную переменную.

Коммуникация между утилитами

Коммунакция между этими утилитами осуществялется с помощью POSIX Message Queue.

Первая утилита при запуске создаёт очередь сообщений для приёма запросов на предоставление статистики. В запросе содержится имя очереди сообщений, в которое надо отправить ответ. Имя очереди сообщений генерируется по подаваемым на вход параметрам и имеет следующий формат:

```
\mq-udpsniff_<if name>_<src ip>_<src port>_<dest ip>_<dest port>
```

Второй утилите на вход подаётся такое имя очереди сообщений, чтобы отправить запрос на получение статистики. Перед отправкой запроса создаётся очередь сообщений для получения ответа и её имя помещается в сообщение запроса.

Сборка

Сборка из исходных файлов

В этом проекте используется система сборки GNU Autotools, поэтому необходимо установить пакет autoconf:

```
apt-get install autoconf
```

Для сборки необходимо из директории с исходным кодом запустить следующие команды:

```
$ autoreconf --install
$ mkdir build && cd build
$ ../configure --prefix /path/to/install
$ make install
```

После выполнения этих команд в каталоге /path/to/install/bin (или если команда `configure` была выполнена без опции prefix - в стандартных директориях /usr/bin или /usr/local/bin) будет создано два исполняемых файла: `udp-sniff` и `print-stats`.

Сборка из архива

Для сборки необходимо из каталога с распакованным исходным кодом запустить следующие команды:

```
$ ./configure --prefix /path/to/install
$ make install
```

Будут созданы те же файлы, что и при сборке из исходных файлов.

Запуск утилит

Привилегии udp-sniff

В утилите `udp-sniff` для прослушивания трафика на интерфейсе используется Raw Socket, требующий привилегии CAP_NET_RAW. Чтобы её установить, необходимо выполнить команду:

```
sudo setcap cap_net_raw+ep ./udp-sniff
```

Запуск udp-sniff

Утилита 'udp-sniff' принимает на вход до 6 аргументов:

- имя интерфейса,
- опциональный: вариант исполнения программы (см. раздел "Описание. Варианты запуска"). По умолчанию - второй вариант.
- 4 опциональных (отвечающих за фильтрацию пакетов): IP-адрес источника, IP-адрес назначения, порт источника, порт назначения. Если опция не указана, то для соответствующего параметра устанавливается значение по умолчанию `0`. Это значение указывает, что любой ip адрес (или порт) соответсвующего параметра пакета будет учитываться в статистике.

```
$ udp-sniff lo --dest-ip 127.0.0.1 --dest-port 1234
```

Запуск print-stats

Утилита 'print-stats' принимает на вход 1 аргумент - имя очереди сообщений для отправки запроса.

```
$ print-stats /mq-udpsniff_lo_0_0_127.0.0.1_1234
Device name: lo
-----------------------------------------
source ip       sport dest ip         dport
0               0     127.0.0.1       1234
-----------------------------------------
packets 0 bytes 0
```

(Чтобы проверить доступные варианты имён очередей сообщений можно запустить команду `ls /dev/mqueue/mq-udpsniff*`).

Подача трафика на интерфейс

TBD

```
```

```
nc -u -l 127.0.0.1 1234 > /dev/null
nc -u 127.0.0.1 1234 < /dev/urandom
```

Результаты профилироания

TBD

Авторство и лицензия

Автор

Лицензия

udp_sniff_SOURCES =
udpsniff/main.c
common.h
netinet_helper.h
netinet_helper.c
udpsniff/control.h
udpsniff/control.c
udpsniff/exec_option/exec_options.h
udpsniff/exec_options/exec_option.c
udpsniff/exec_options/priv_exec_options.h
udpsniff/exec_options/exec_option1.c
udpsniff/exec_options/exec_option2.c
mq_common.c
udpsniff/mq_interface.c