Прикладной уровень

Прикладной уровень обеспечивает инструменты для выполнения прикладных задач. Он помогает обеспечить передачу данных при помощи лежащих ниже протоколов транспортного и сетевого уровней между хостами по модели «клиент — сервер» или между равноправными участниками взаимодействия.

В оригинальной спецификации (документ Request for Comments) стека протоколов для интернета были перечислены следующие протоколы прикладного уровня, которые обеспечивали основные потребности юного интернета:

1. Удалённое подключение к хостам: Telnet.
2. Передача файлов: File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP).
3. Передача электронных писем: Simple Mail Transfer Protocol (SMTP).
4. Поддержка сети: Domain Name System (DNS).
5. Инициализация хоста: BOOTP.
6. Удалённое управление хостом: Simple Network Management Protocol (SNMP), Common Management Information Protocol over TCP (CMOT).

Примеры прикладных протоколов:

- BGP — роутинг и обмен информацией в интернете.
- DHCP — автоматическая настройка IP-адреса хоста.
- DNS — разрешение доменных имён.
- FTP — передача файлов.
- HTTP — уже знакомый нам протокол, на котором построен весь современный веб.
- HTTPS — защищённая версия HTTP.
- LDAP — протокол, обеспечивающий доступ к службам каталогов.
- MQTT — протокол, используемый для передачи телеметрии между промышленными устройствами.
- NTP, PTP — протоколы, обеспечивающие синхронизацию часов.
- SIP — протокол, используемый для телефонии.
- SMTP — обмен электронными письмами.
- SNMP — сбор и организация информации об устройствах в сети.
- SSH — обеспечение удалённого доступа.
- XMPP — обмен сообщениями.

Программа Wireshark

Для того чтобы заглянуть внутрь сетевых пакетов, есть программа Wireshark. Она позволяет «перехватывать» сетевой трафик и отображать служебную информацию и данные пакетов, которые были обработаны хостом.

Для начала давайте установим программу:

```
sudo apt-get install wireshark
```

Теперь давайте запустим её с правами суперпользователя:

```
sudo wireshark
```

Включим перехват трафика на нашем интерфейсе.

Интерфейс Wireshark содержит пять основных областей:

- Командные меню представляют собой стандартные раскрывающиеся меню, расположенные вверху окна.
- Меню File («Файл») предназначено для сохранения захваченных пакетов, для открытия файла с уже сохранёнными данными пакетов, а также для выхода из программы. Команды в меню Capture («Захват») позволяют начать захват пакетов.
- Окно списка пакетов отображает построчно информацию по каждому захваченному пакету, включая номер пакета, время, когда пакет был перехвачен, адреса источника и приёмника, тип протокола, а также специальную информацию, относящуюся к протоколу.
- В окне деталей заголовка пакета отображается подробная информация о пакете, выбранном в предыдущем окне (строка с этим пакетом подсвечена).
- Окно содержимого пакета отображает всё, что содержится в захваченном пакете, в шестнадцатеричном формате и в формате ASCII.

Создадим немного сетевой активности:

```
curl http://ya.ru
```

Здесь мы можем увидеть DNS-запросы (чтобы узнать, какому IP-адресу или адресам соответствует доменное имя), установку TCP-соединения (тот самый хендшейк, о котором мы говорили), затем HTTP-запрос и HTTP-ответ (здесь мы можем увидеть уже знакомые нам заголовки).

Если мы сделаем запрос:

```
curl https://ya.ru
```

увидим установку TCP-соединения, защищённого соединения, а вот содержимое ответа будет в зашифрованном виде.

Чтобы остановить процесс захвата пакетов, нажмите на кнопку «Стоп». Далее вы можете сохранить захваченный трафик (или при попытке возобновить захват программа вас спросит об этом сама) или начать новый захват.

Основные фильтры

Фильтр по протоколу: достаточно в строке фильтра ввести название протокола и нажать «Ввод». На экране останутся пакеты, которые относятся к искомому протоколу. Фильтр выглядит так:

```
http, arp, tls
```

Фильтр по IP-адресу: в зависимости от направления трафика фильтр будет немного отличаться. Например, мы хотим отфильтровать по IP-адресу отправителя:

```
ip.src== 192.168.88.223
```

В случае если нам нужно исключить какой-то адрес из поля отбора, то необходимо добавить !=. Пример:

```
ip.src!=80.68.246.17
```

Фильтр по получателю будет выглядеть так — ip.dst == x.x.x.x, а если хотим увидеть пакеты вне зависимости от направления трафика, то достаточно ввести:

```
ip.addr==50.116.24.50
```

Фильтр по номеру порта: при анализе трафика мы можем настроить фильтр по номеру порта, по которому осуществляет передачу трафика тот или иной протокол. Номера всех зарегистрированных портов можно узнать здесь. Пример:

```
ftp.port==21
```