
давайте изучим
tcpdump
с Джулией Эванс
О чем говорят эти компьютеры?
Используй tcpdump и узнаешь!
TCP
UDP

Што здесь?

Мужская страница tcpdump'а начинается вот так:

```
NAME
       tcpdump - dump traffic on a network

SYNOPSIS
       tcpdump [ -AbdDefhHIJKlLnNOpqStuUvxX# ] [ -B buffer_size ]
               [ -c count ]
               [ -C file_size ] [ -G rotate_seconds ] [ -F file ]
               [ -i interface ] [ -j tstamp_type ] [ -m module ] [ -M secret
               [ --number ] [ -Q in|out|inout ]
               [ -r file ] [ -V file ] [ -s snaplen ] [ -T type ] [ -w file ]
               [ -W filecount ]
               [ -E spi@ipaddr algo:secret,...  ]
               [ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]
               [ --time-stamp-precision=tstamp_precision ]
               [ --immediate-mode ] [ --version ]
               [ expression ]
```

Джулия Эванс

@b0rk

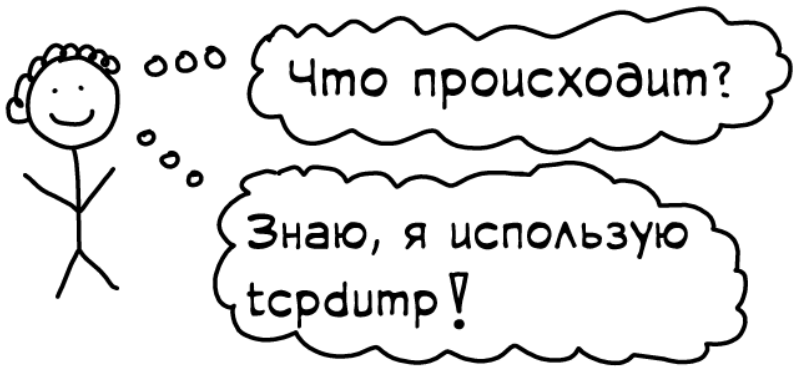
http://jvns.ca

ее блог

Для чего нужен tcpdump?

Tcpdump захватывает и выводит сетевой трафик.

Например! Вчера с моего ноута очень медленно уходили DNS-запросы.




```
$ sudo tcpdump -n -i any port 53
```

DNS-запросы

```
10:52:03.992138 IP 192.168.1.241.63019 > 192.168.1.1.53: 44000+ A? ask.metafilter.com. (36)
10:52:08.972719 IP 192.168.1.241.63019 > 192.168.1.1.53: 44000+ A? ask.metafilter.com. (36)
10:52:13.919782 IP 192.168.1.241.63019 > 192.168.1.1.53: 44000+ A? ask.metafilter.com. (36)
10:52:13.928894 IP 192.168.1.1.53 > 192.168.1.241.63019: 44000 2/0/0 CNAME metafilter.com.,
A 54.186.13.33 (80)
```

ответ DNS

Я вижу, что было 3 DNS-запроса (в 10:52:03, 10:52:08, 10:52:13), но ответ пришел только на последний!

Я решила, что проблема кроется в моем роутере. После его перезагрузки интернет заработал нормально!

Давайте научимся дебажить проблемы с tcpdump'ом.

Вопросы, на которые ответит tcpdump

→ Какие DNS-запросы отправляет мой лэптоп?

"tcpdump -i any port 53"

→ У меня на 1337 порту работает сервер. Доходят ли туда пакеты???

"tcpdump -i any port 1337"

→ Какие пакеты приходят на мой сервер от IP 1.2.3.4?

"tcpdump port 1337 and host 1.2.3.4"

→ Покажи мне все неудачные DNS-запросы

"tcpdump udp[11] &0xf==3"

(Сложно, но это работает!)

→ Как долго на этой машине остаются открытыми TCP соединения?

"tcpdump -w packets.pcap"

а потом проанализировать packets.pcap в Wireshark

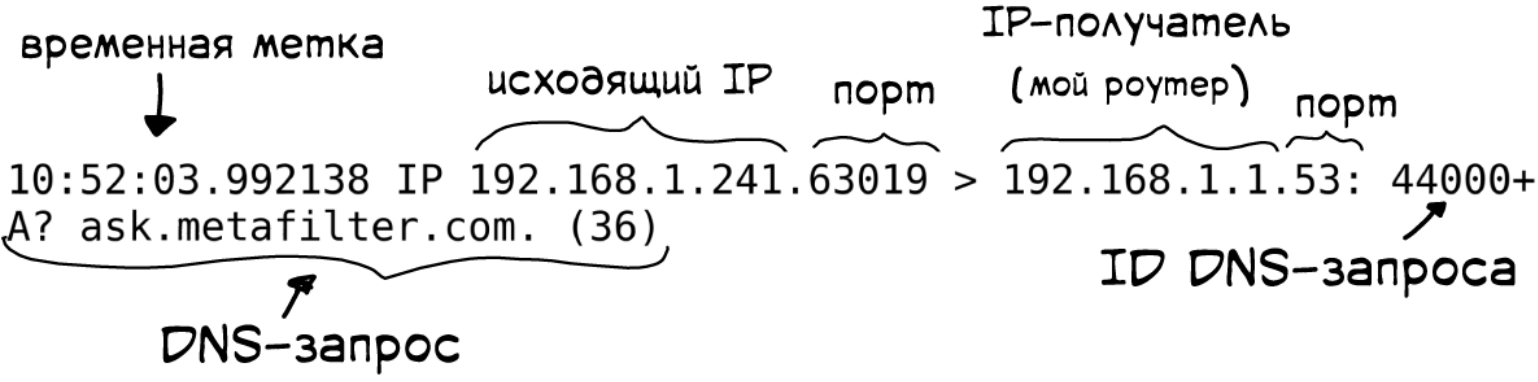
Как читать вывод tcpdump'а

Каждая строка tcpdump'а представляет из себя пакет.

Обычно я обращаю внимание на следующее:

- ★ IP-адрес и порт источника и получателя
- ★ временную метку
- ★ TCP-флаги (легко увидеть начало TCP-соединения)
- ★ DNS-запрос, для пакетов DNS
- ★ и это все!

UDP-пакет:



TCP-пакет:

TCP-флаги
"." значит ACK

```
11:36:26.353797 IP 192.168.1.241.45296 > 192.241.182.146.443: Flags [.],
ack 2291349910, win 319, options [nop,nop,TS val 10967552 ecr 580196754],
 length 0
```

Видели когда-нибудь ошибку "Connection refused"?

Вот как она выглядит в tcpdump'е!

SYN

```
12:16:38.944390 IP6 localhost.48680 > localhost.8999: Flags [S]
12:16:38.944458 IP6 localhost.8999 > localhost.48680: Flags [R.]
```

RST ACK

Чтобы открыть соединение мы отправили SYN, но сервер ответил пакетом "RST". Это преобразуется в "connection refused".

Фильтры BPF
(самый маленький гайд)

Julia Evans
@b0rk

В tcpdump используется маленький язык под названием BPF, чтобы мы могли фильтровать пакеты.

port 53

проверяет, является ли исходящий ИЛИ входящий порт 53м. Соответствует TCP порту 53 и UDP порту 53.

host 192.168.3.2

проверяет, является ли исходящий или входящий IP 192.168.3.2

host 11.22.33.44 and port 80

можно использовать 'and' (и), 'or' (или) и 'not' (не).

→ src port 80
→ dest port 80
→ tcp port 80

это именно то, чем кажется :)

также как и

src host 1.2.3.4
dest host 1.2.3.4

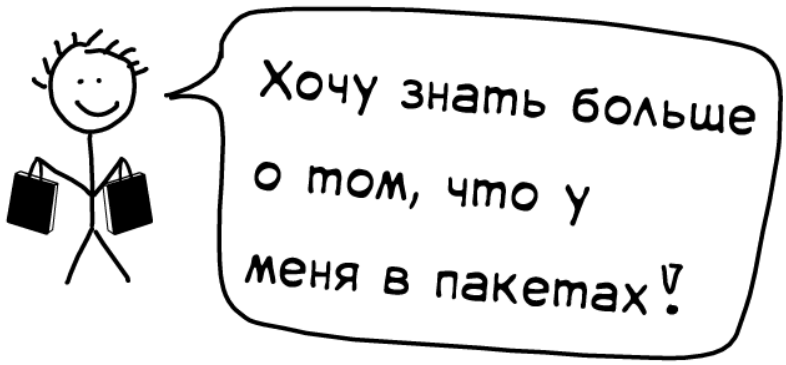
udp[11] & 0xf == 3

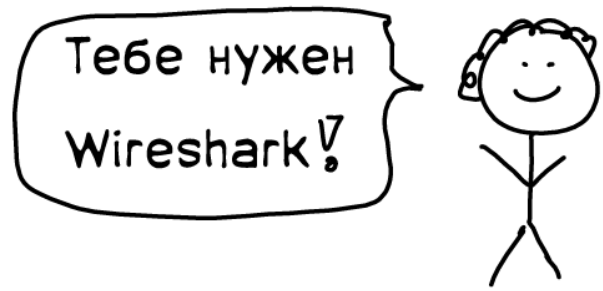
можно использовать битовые вычисления на содержимом пакета. Это, например, проверяет на ответ DNS "NXDOMAIN"!

(пришлось погуглить, чтобы найти, но оно работает! ☺)

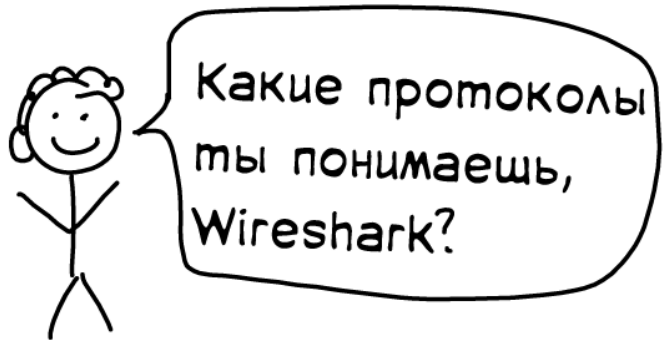
Это не все, но до сих пор мне было этого достаточно! ☺

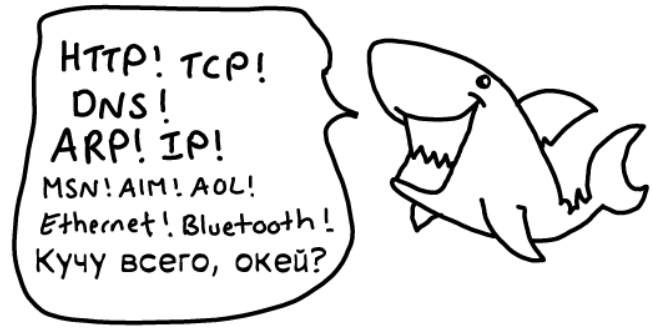
♡ Wireshark ♡





Wireshark – невероятно мощная утилита для анализа пакетов!





Штуковины в Wireshark:

- ★ приятный графический интерфейс!
- ★ может склеивать TCP-пакеты из одного соединения!
- ★ простой поиск по пакетам!

Если нужно анализировать пакеты из tcpdump'а в Wireshark, можно:

1. сохранить файл .pcap и открыть его в Wireshark
2. использовать это заклинание и законвейерить вывод tcpdump'а в Wireshark!

```
ssh some.remote.host tcpdump -pni any -w - -s0 -U port 8888
  | wireshark -k -i -
```

мои ♡ любимые ♡ аргументы командной строки

Эти три аргумента я использую чаще всего:

для интерфейсов

На каком сетевом интерфейсе захватывать пакеты. Я часто использую `-i any`. Вам не всегда нужен дефолтовый интерфейс, который выбирает tcpdump.

Например: sudo tcpdump -i lo покажет пакеты на локальном "loopback" интерфейсе

для записи

Вместо вывода пакетов запишите их в файл! ОЧЕНЬ полезно для последующего анализа. Я пользуюсь этим постоянно!

Например: sudo tcpdump host 8.8.8.8 -w my_packets.pcap сохраняет пакеты к/от 8.8.8.8 в файл.

для подсчета

Записывая пакеты в файл, надо проявлять бдительность, ведь можно забить свой накопитель до отказа. `-c 10000` ограничит захват 10000 пакетов.

Например: sudo tcpdump -c 1000 -w my_packets.pcap dest port 8080

вот еще несколько крутых аргументов:

Выводит содержимое пакета!
Например, у меня есть веб-сервер на 7777 порту.

```
$ sudo tcpdump -A dest port 7777
```

покажет мне все HTTP-запросы, отправляемые на этот сервер. Работает только для HTTP, но не HTTPS.

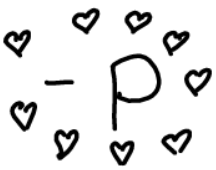
(Правда, мне больше нравится ngrep, чем tcpdump -A для просмотра тела HTTP-запросов)

По умолчанию tcpdump преобразует IP-адреса в имена хостов. -n заставляет его всегда выводить IP-адреса.

для Ethernet

Добавляет данные Ethernet. Показывает MAC-адрес с которого пришел пакет.

Например: sudo tcpdump -e -i any port 443

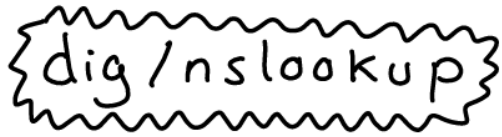


Ограничиваем пакеты теми, что приходят/уходят с нашего компьютера.

Штуки для сетевого администрирования

Наконец, есть много других утилит кроме tcpdump!
Не будем подробно останавливаться, но вот список!

"Между этими компьютерами вообще есть соединение?"



"Существует ли этот домен?"

"Использую ли я этот порт?"

"Какой у меня ip-адрес?"

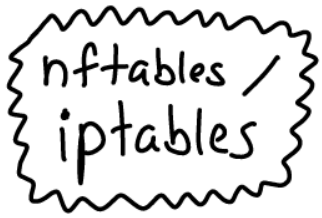
Настраивает интерфейсы, маршруты и всякое. Наследник ifconfig'а.

Лицезрейте свою таблицу ARP!

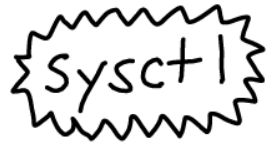
grep для сети

Через какие серверы проходит маршрут вот до того сервера?

netcat! открывайте TCP-соединения вручную.



Настраиваем файерволы и NAT!



Настраиваем размер буфера сокетов и не только!

Для понимания своих езернетовых соединений.

nmap

Сканируем порты в своей сети.

whois

Смотрим информацию о домене.

lsof

Какие порты используются?

telnet

Узнаем, открыт ли порт n на сервере.

ssh

это просто нельзя забыть

network manager

GUI-утилита для настройки сети на вашем лаптопе :)

nethogs/ab/nload
iptraf/netperf/iperf
iftop/netsniff-ng

Куча утилит для оценки производительности/бенчмаркинга (каждая делает что-то свое).

paping

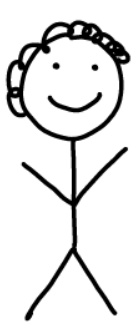
ping, только по TCP.

OpenVpn

настраиваем VPN!

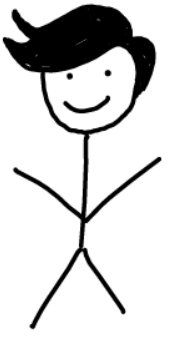
socat

Как netcat, но больше фич.



Огромное спасибо за чтение!

Теперь, когда я знаю основы, мужская страница не выглядит такой страшной!



Понравилось?

Читайте больше на:
http://jvns.ca/zines

Перевела команда FirstVDS.ru