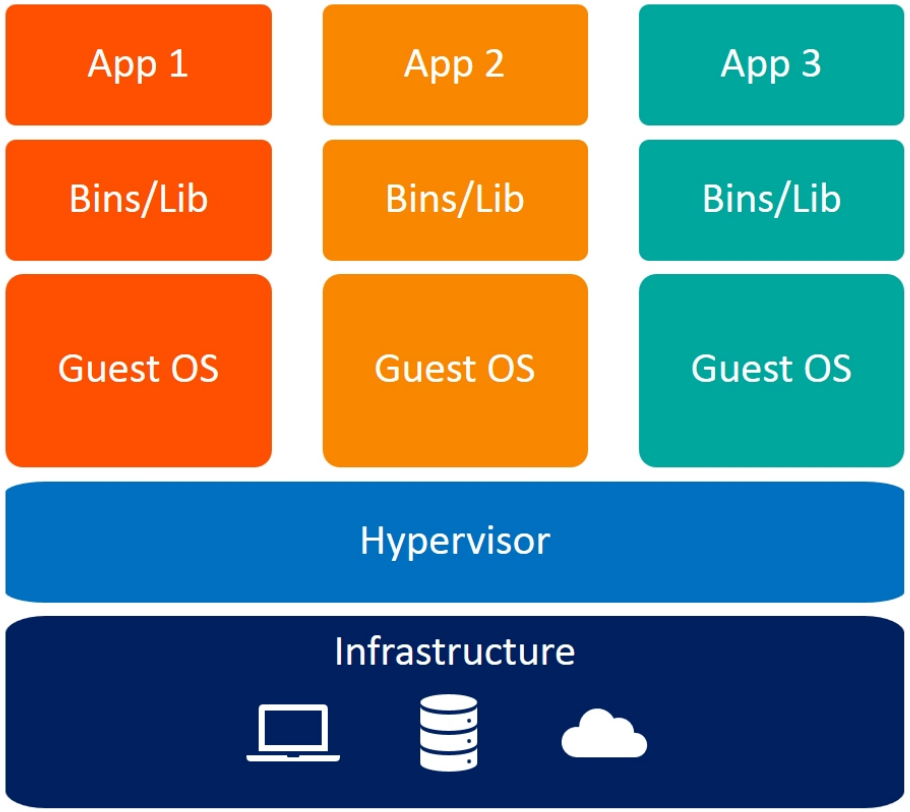
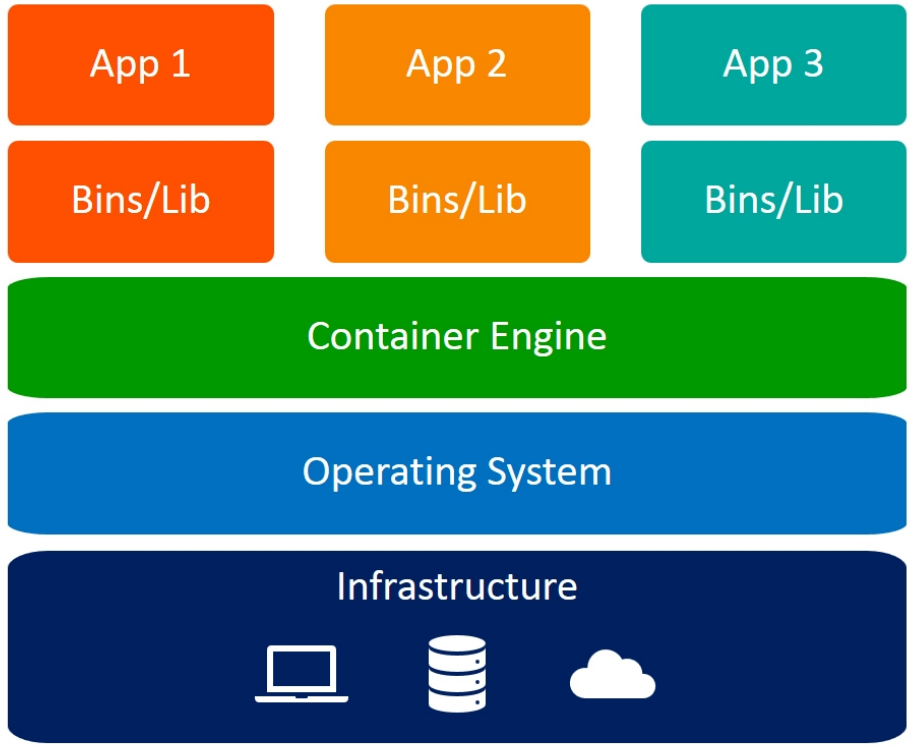
АКОС 13

BPF & seccomp

Виртуализация vs Контейнеризация



Virtual Machines



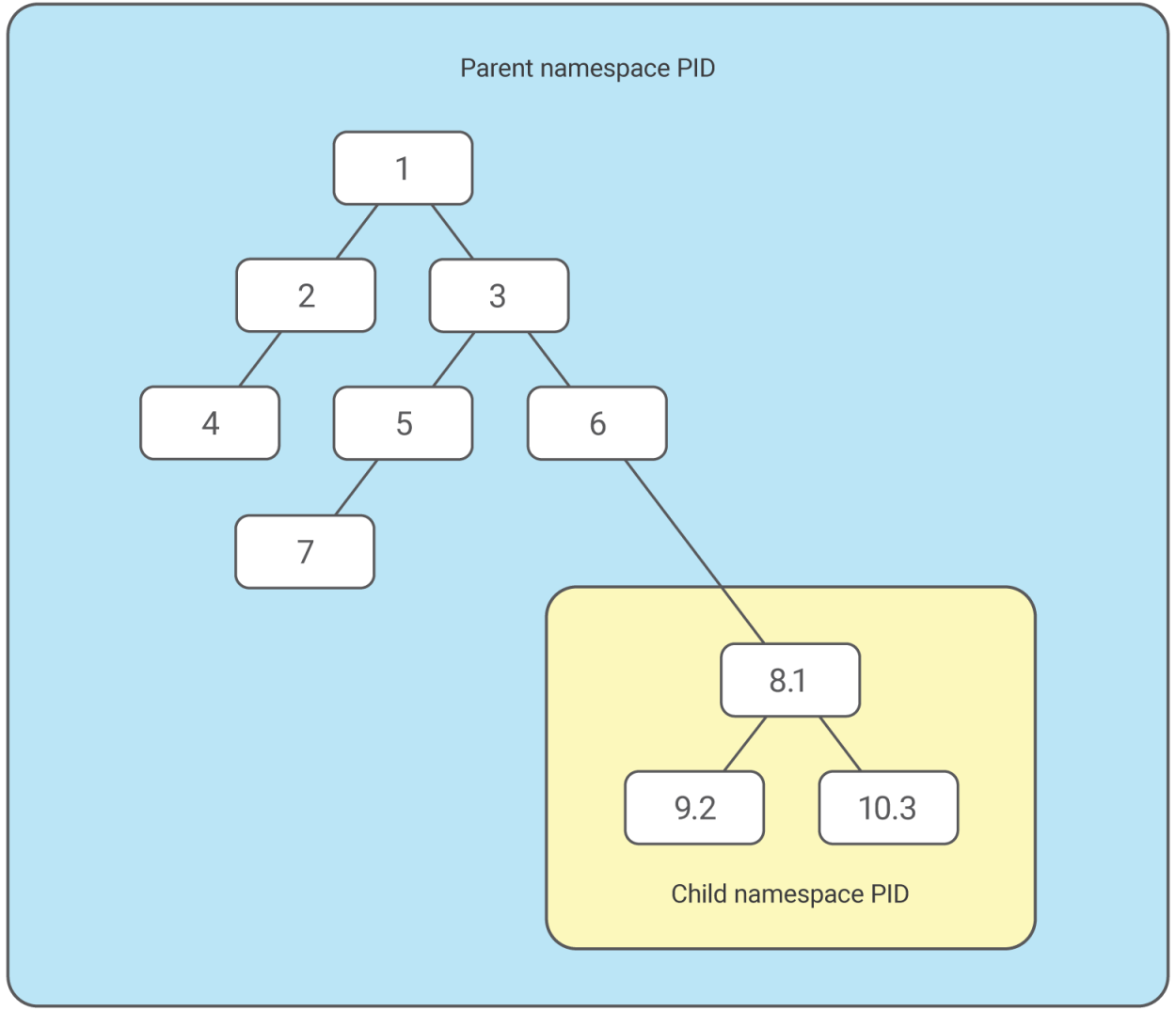
Containers

chroot

- Позволяет поменять корень директории
- $ sudo chroot ./fs – поменять корень на fs
- sudo debootstrap --variant=minbase stable ./fs http://deb.debian.org/debian - подготовить файловую систему

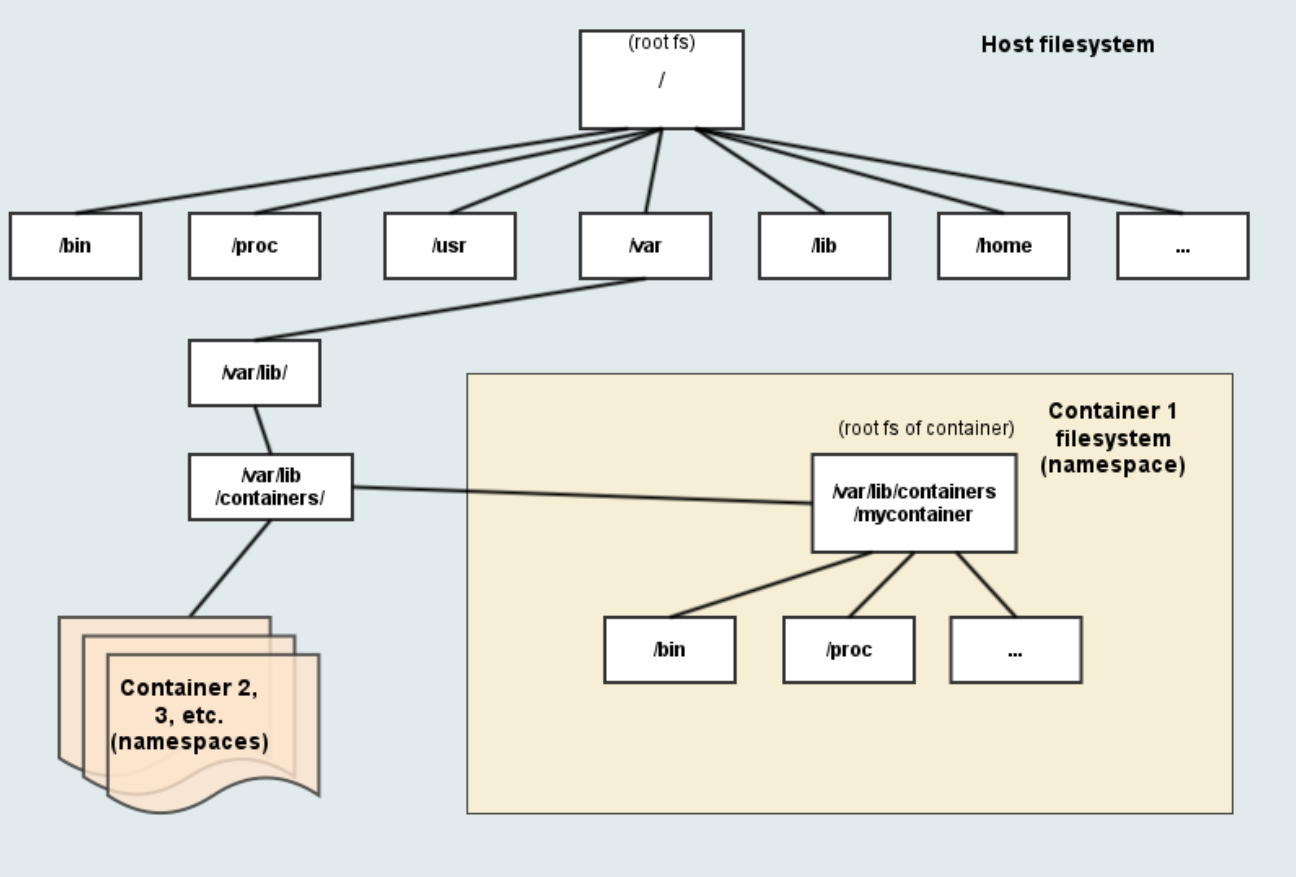
- mount –t proc proc /proc
- ls /proc -l

Namespaces



$ docker run -it --rm busybox

Mount



- https://docs.docker.com/storage/storagedriver/overlayfs-driver/

CSPF (CMU / Stanford Packet Filter)

- Фильтр для пакетов
- Происходит в пространстве ядра, а не пользователя
- Проблема:

CSPF (CMU / Stanford Packet Filter)

- Фильтр для пакетов
- Происходит в пространстве ядра, а не пользователя
- Проблема:
  - Слишком высокие права при работе в ядре

CSPF (CMU / Stanford Packet Filter)

- Фильтр для пакетов
- Происходит в пространстве ядра, а не пользователя
- Проблема:

  Слишком высокие права при работе в ядре
- Решение:

  Виртуальная машина

BPF (Berkeley Packet Filters)

- Виртуальная машина BPF:
  - Два доступных для пользователя 32-битных регистра
  - 64 байта памяти
  - Инструкции перехода (но только вперёд)

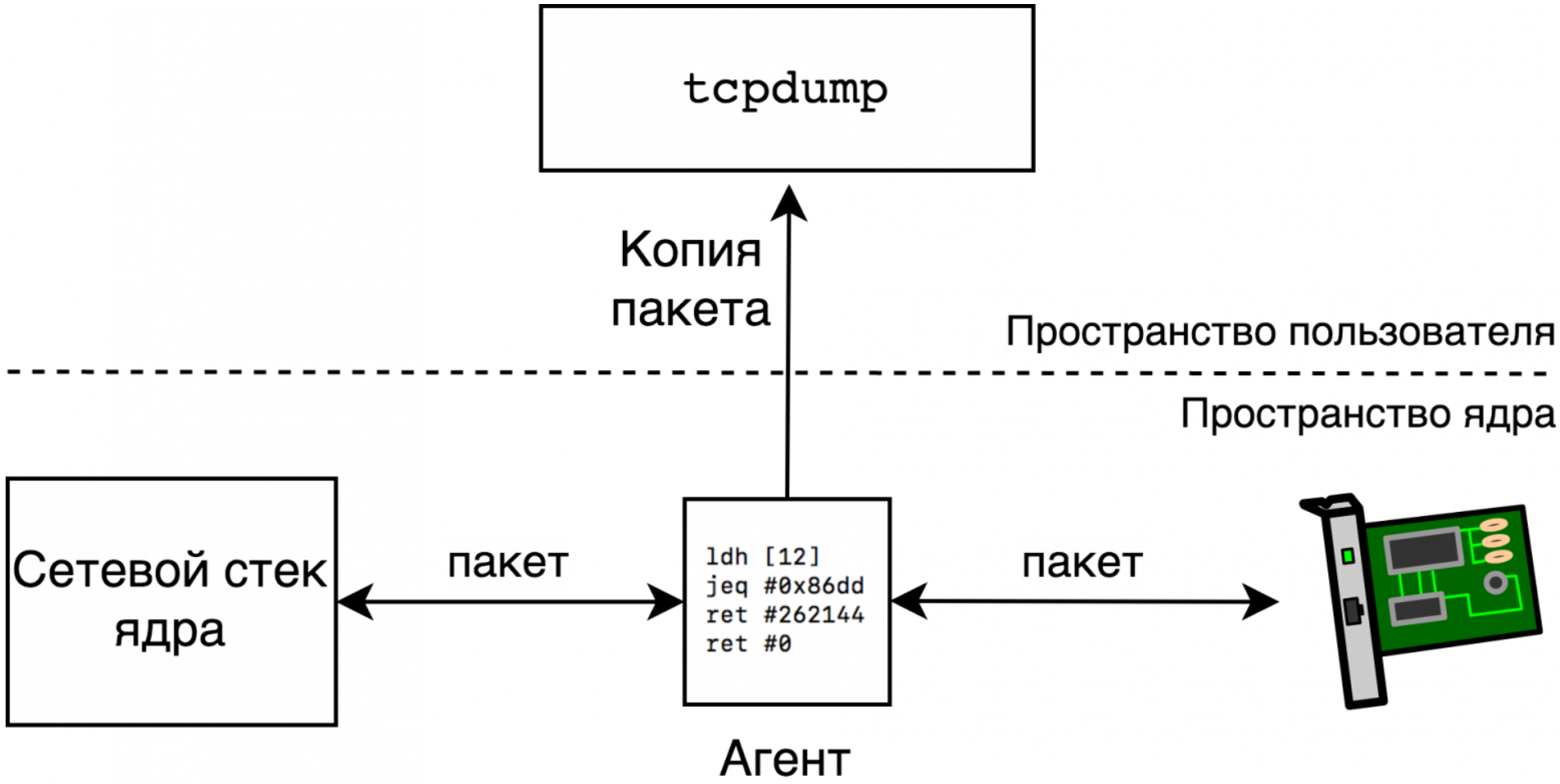
BPF (Berkeley Packet Filters)

- Виртуальная машина BPF:
  - Два доступных для пользователя 32-битных регистра
  - 64 байта памяти
  - Инструкции перехода (но только вперёд, поэтому нет циклов)

Схема запуска

1. Пользователь создает программу для архитектуры BPF
2. При помощи системного вызова загружает её в ядро
3. Подключает её к генератору событий в ядре (например, получение нового пакета на сетевой карте)
4. При возникновении события ядро запускает программу (в интерпретаторе)
5. Память программы соответствует региону памяти ядра (например, данным пакета)

tcpdump


tcpdump
Копия пакета
Пространство пользователя
Пространство ядра
Сетевой стек ядра
пакет
ldh [12]
jeq #0x86dd
ret #262144
ret #0
пакет
Агент

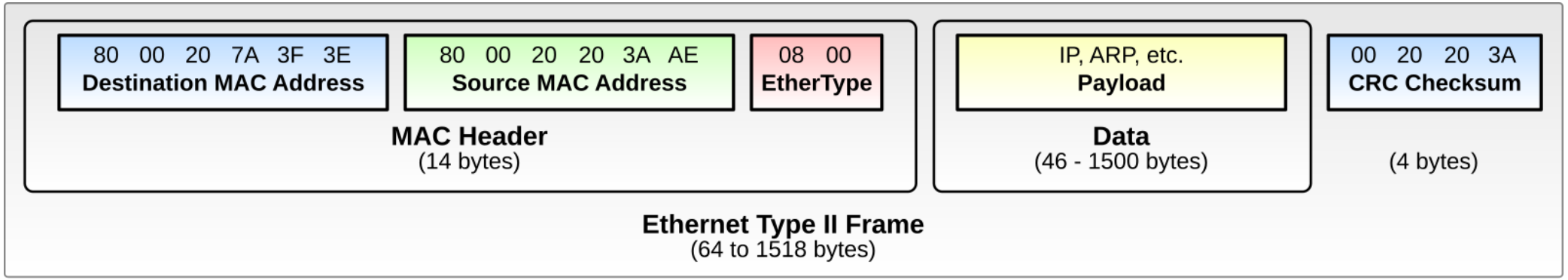
tcpdump

- Загружает в ядро BPF фильтр
- ip a – посмотреть список доступных интерфейсов
- route – посмотреть, через какие интерфейсы идут пакеты
- sudo tcpdump –i eth0 ip – посмотреть пакеты на интерфейсе
- sudo tcpdump –i eth0 ip **–d** ip – можно смотреть на байткод
- sudo strace -f -e trace=%network tcpdump -p -i eth0 ip – посмотреть на загрузку пакеты

Байт-код фильтра

- sudo tcpdump –i eth0 ip **–d** ip

```
(000) ldh      [12]
(001) jeq      #0x800           jt 2    jf 3
(002) ret      #262144
(003) ret      #0
```



eBPF

Изменения по сравнению с BPF:

- JIT-компилятор
- Фильтры для seccomp
- Модуль xt_bpf позволяет писать правила для iptables
- Модуль cls_bpf для написания классификаторов трафика

Программируем BPF

- Формат инструкций:

```
   16     8     8      32
| code | jt | jf |  k   |
```

- На С одна команда записывается в структуре:

```
struct sock_filter {
        __u16   code;
        __u8    jt;
        __u8    jf;
        __u32   k;
}
```

Программируем BPF

- Вся программа записывается в виде структуры:

```
struct sock_fprog {
        unsigned short len;
        struct sock_filter *filter;
}
```

Программируем BPF

- Можно использовать макросы из <linux/filter.h> или писать на *ассемблере*

```
struct sock_filter code[] = {
        { 0x28, 0, 0, 0x0000000c },
        { 0x15, 0, 1, 0x000086dd },
        { 0x06, 0, 0, 0x00040000 },
        { 0x06, 0, 0, 0x00000000 },
};
struct sock_fprog prog = {
        .len = ARRAY_SIZE(code),
        .filter = code,
};
```

```
struct sock_filter code[] = {
        BPF_STMT(BPF_LD|BPF_H|BPF_ABS, 12),
        BPF_JUMP(BPF_JMP|BPF_JEQ|BPF_K, ETH_P_IPV6, 0, 1),
        BPF_STMT(BPF_RET|BPF_K, 0x00040000),
        BPF_STMT(BPF_RET|BPF_K, 0),
}
```

Seccomp + BPF

- Можно писать фильтры для системных вызовов
- Удобно писать с помощью библиотеки libseccomp
- $ sudo apt-get install libseccomp-dev

1. seccomp_init – какое действие выполнять
   - SCMP_ACT_KILL – убить процесс
   - SCMP_ACT_TRAP – перехватить сискол, поток кинет сигнал SIGSYS
2. seccomp_rule_add(ctx, SCMP_ACT_ALLOW, SCMP_SYS(read), 0); - добавляем правила
3. seccomp_load(ctx); - активируем правила

Seccomp и Docker

- Docker использует seccomp, чтобы запретить некоторые сисколы (например, reboot)
- Можно при запуске контейнера указать ограничения на сисколы:
  - $ docker run **--security-opt seccomp:chmod.json** busybox chmod 400 /etc/hostname

Ресурсы

- Оригинальная статья про BPF
  - “The BSD Packet Filter: A New Architecture for User-level Packet Capture"
- Серия статей про BPF на хабре
  - BPF для самых маленьких

- Статья на про seccomp

Docker

- Installation