?????????? ??????

- Ipfw

Ipfw

Когда-то давно мы настраивали фаервол в Linux с помощью iptables. При этом отмечалось, что утилиту iptables я нахожу исключительно неудобной по сравнению с FreeBSD'шным ipfw. Сегодня мы наконец-то познакомимся с этим ipfw и постараемся ответить на вопрос, действительно ли он удобнее. Отмечу, что на момент написания этих строк FreeBSD предлагает аж *три* фаервола на выбор — ipfw, pf и ipf. Однако по моим представлениям из них ipfw используется чаще всего.

Описанные в этом посте действия были проверены мной на FreeBSD 10.3, но по идее не должны ничем отличаться в других версиях FreeBSD.

Итак, первое, что требуется выполнить при настройке ipfw — это команду:

```
sudo kenv net.inet.ip.fw.default_to_accept=1
```

По умолчанию ipfw запрещает передачу вообще всех пакетов. Если вы сидите на сервере по s

Теперь включаем ipfw:

```
sudo kldload ipfw
```

При этом в /var/log/messages мы должны увидеть что-то вроде:

```
ipfw2 (+ipv6) initialized, divert loadable, nat loadable,
  default to accept, logging disabled
```

Как видите, ipfw представляет собой обычный модуль ядра.

Давайте посмотрим на текущий список правил:

```
sudo ipfw list
```

Должны увидеть:

```
65535 allow ip from any to any
```

Приведенный синтаксис довольно очевидный — номер правила, резрешить / запретить, назва

Также можно посмотреть время, когда правило последний раз срабатывало:

```
sudo ipfw -t list
```

Давайте попробуем добавить парочку правил. Например, запретим пинги:

```
# флаг -q выполняет команду в "тихом" режиме, без вывода
sudo ipfw -q add 00100 deny icmp from any to any
sudo ipfw list
```

Как видите, указывая номер правила, мы знаем точно, на какую позицию оно встанет. Принци

Попытаемся что-нибудь пинганути и убедимся, что пинги больше не ходят. Теперь, когда мы осознали свою ошибку, удаляем правило:

```
sudo ipfw -q delete 00100
```

Заметьте, что ipfw позволяет создавать множество правил с одинаковыми номерами. В этом с

```
sudo ipfw -q -f flush
```

Перед тем, как реально добавлять правило, не лишним бывает собрать статистику о том, как

```
sudo ipfw -q add 00100 count udp from me to any out via em0
sudo ipfw -q add 00200 count udp from any to me in via em0
```

Как видите, тут демонстрируется не только новое действие (count), но и расширенный синта *продолжают* применяться.

Теперь посмотрим количество пакетов с их суммарным размером в байтах, соответствующих правилу:

```
sudo ipfw -a list
# или:
sudo ipfw show
```

Счетчики можно при желании обнулить так:

```
# для конкретного правила
sudo ipfw zero 00200

# для всех правил
sudo ipfw zero
```

В правилах можно использовать не только слова `me` и `any`
, но и указывать конкретные хосты или подсети:

```
sudo ipfw -q add 00100 deny tcp from any to 192.168.0.1
sudo ipfw -q add 00200 deny tcp from any to 192.168.0.1/24
```

Также можно указывать порты или диапазоны портов:

```
sudo ipfw -q add 00100 deny tcp from any to 192.168.0.1/24 80
sudo ipfw -q add 00200 deny tcp from any to 192.168.0.1/24 1-1024
```

Помимо сбора статистики о срабатывании правил, их также можно логировать:

```
sudo ipfw enable verbose
```

Пример добавления логируемых правил:

```
sudo ipfw -q add 00200 deny log \
  tcp from any to 212.193.240.1/24 80

sudo ipfw -q add 00200 deny log logamount 3 \
  tcp from any to 212.193.240.1/24 80
```

Если теперь несколько раз попытаться прителнетиться к хосту из используемой выше сети и

```
ipfw: 200 Deny TCP 10.0.2.15:41738 212.193.240.242:80 out via em0
ipfw: 200 Deny TCP 10.0.2.15:43564 212.193.240.242:80 out via em0
ipfw: 200 Deny TCP 10.0.2.15:55804 212.193.240.242:80 out via em0
ipfw: limit 3 reached on entry 200
```

Логирование можно глобально включать и выключать таким образом:

```
sudo sysctl net.inet.ip.fw.verbose=1
sudo sysctl net.inet.ip.fw.verbose=0
```

Если в правиле не указывается logamount, используется следующее значение:

```
sudo sysctl net.inet.ip.fw.verbose_limit=5
```

Значение по умолчанию равно нулю, что означает отсутствие ограничений.

Сбросить счетчики логирования можно так:

```
sudo ipfw resetlog
```

```
Для сохранения правил при перезагрузке системы создадим файл /etc/ipfw.rules следующего
содержания:
#!/bin/sh

fwcmd="ipfw -q add"

ipfw -q -f flush
```

```
$fwcmd 00100 deny log tcp from any to 1.2.3.0/24
$fwcmd 00200 deny log tcp from any to 4.5.6.0/24
$fwcmd 65000 allow ip from any to any
```

Важно!
Если хотите подключиться к машине после перезагрузки, обязательно добавьте последнее п
`net.inet.ip.fw.default_to_accept`
. Однако после ребута вернется поведение фаервола по умолчанию, при котором запрещаетс

Заметьте, что только что мы написали обыкновенный шелл скрипт. Следовательно, в нем можно использовать сокращения имен команд, как продемонстрировано выше, а также условные операторы, процедуры и вот это все. Проверяем скрипт:

```
sudo sh /etc/ipfw.rules
sudo ipfw list
```

Если все OK, в /etc/rc.conf дописываем:

```
firewall_enable="YES"
firewall_script="/etc/ipfw.rules"
firewall_logging="YES"
```

Если решили включить логирование, убедитесь, что вы не забыли дописать в /etc/sysctl.conf ч

```
net.inet.ip.fw.verbose_limit=5
```

Проверяем:

```
sudo kldunload ipfw
sudo service ipfw start
sudo ipfw list
```

Если все было сделано правильно, долждны увидеть наши правила. Кроме того, они должны
Это все, о чем я хотел рассказать. Подробности вы найдете в `man 8 ipfw`. Маны во FreeBSD, как всегда, классные. В частности, из мана вы узнаете про такие элементы синтаксиса правил ipfw, оставшиеся за рамками поста, как skipto, tag / untag и другие.

Неправда ли, синтаксис в ipfw куда более читаемый, чем вот эти все `-A`, `-p`, `-j`, `--sport` и `-m multiport` в iptables?

?????? ????????????? ??????????

- Настраиваем фаервол во FreeBSD при помощи ipfw