Zapret Настройка BSD-подобных систем

Поддерживаемые версии

FreeBSD 11.x+ , **OpenBSD** 6.x+, частично **MacOS Sierra** +

Caution

На более старых может собираться, может не собираться, может работать или не работать. На **FreeBSD** 10 собирается и работает `dvtws`. С `tpws` есть проблемы из-за слишком старой версии компилятора clang. Вероятно, будет работать, если обновить компилятор. Возможна прикрутка к последним версиям pfsense без веб интерфейса в ручном режиме через консоль.

Особенности BSD систем

Отсутствие nfqueue

В **BSD** нет `nfqueue`. Похожий механизм - divert sockets. Из каталога `nfq/` под **BSD** собирается `dvtws` вместо `nfqws`. Он разделяет с `nfqws` большую часть кода и почти совпадает по параметрам командной строки.

Типы Firewall

FreeBSD содержит 3 фаервола : **IPFilter**, **ipfw** и **Packet Filter (PF в дальнейшем)**. **OpenBSD** содержит только **PF**.

Сборка

Под **FreeBSD** `tpws` и `dvtws` собираются через `make`.

Под **OpenBSD**:

```
make bsd
```

Под **MacOS**:

```
make mac
```

FreeBSD make распознает BSDmakefile, **OpenBSD** и **MacOS** - нет. Поэтому там используется отдельный target в Makefile. Сборка всех исходников:

```
make -C /opt/zapret
```

Divert сокеты

Divert сокет это внутренний тип сокета ядра **BSD**. Он не привязывается ни к какому сетевому адресу, не участвует в обмене данными через сеть и идентифицируется по номеру порта `1..65535`. Аналогия с номером очереди `NFQUEUE`. На divert сокеты заворачивается трафик посредством правил ipfw или PF. Если в фаерволе есть правило divert, но на divert порту никто не слушает, то пакеты дропаются. Это поведение аналогично правилам `NFQUEUE` без параметра `--queue-bypass`. На **FreeBSD** divert сокеты могут быть только ipv4, хотя на них принимаются и ipv4, и ipv6 фреймы. На **OpenBSD** divert сокеты создаются отдельно для ipv4 и ipv6 и работают только с одной версией `ip` каждый. На **MacOS** похоже, что divert сокеты из ядра вырезаны. См подробнее раздел про **MacOS**. Отсылка в divert сокет работает аналогично отсылке через raw socket на linux. Передается полностью IP фрейм, начиная с ip загловка. Эти особенности учитываются в `dvtws`.

Lookup Tables

Скрипты `ipset/*.sh` при наличии ipfw работают с ipfw lookup tables. Это прямой аналог ipset. lookup tables не разделены на v4 и v6. Они могут содержать v4 и v6 адреса и подсети одновременно. Если ipfw отсутствует, то действие зависит от переменной `LISTS_RELOAD` в config. Если она задана, то выполняется команда из `LISTS_RELOAD`. В противном случае не делается ничего. Если `LISTS_RELOAD=-`, то заполнение таблиц отключается даже при наличии ipfw.

Загрузка ip таблиц из файла

PF может загружать ip таблицы из файла. Чтобы использовать эту возможность следует отключить сжатие gzip для листов через параметр файла config: `GZIP_LISTS=0`.

Отсутствие splice

BSD не содержит системного вызова splice. `tpws` работает через переброску данных в user mode в оба конца. Это медленнее, но не критически. Управление асинхронными сокетами в `tpws` основано на linux-specific механизме epoll. В **BSD** для его эмуляции используется epoll-shim - прослойка для эмуляции epoll на базе kqueue.

mdig и ip2net

mdig и ip2net полностью работоспособны в **BSD**. В них нет ничего системо-зависимого.

FreeBSD

Подгрузка ipdivert

Divert сокеты требуют специального модуля ядра - `ipdivert`.

- Поместите следующие строки в `/boot/loader.conf` (создать, если отсутствует):

```
ipdivert_load="YES"
net.inet.ip.fw.default_to_accept=1
```

`/etc/rc.conf`:

```
firewall_enable="YES"
firewall_script="/etc/rc.firewall.my"
```

`/etc/rc.firewall.my`:

```
$ ipfw -q -f flush
```

Авто-восстановление правил ipfw и работа в фоне

В `/etc/rc.firewall.my` можно дописывать правила ipfw, чтобы они восстанавливались после перезагрузки. Оттуда же можно запускать и демоны zapret, добавив в параметры `--daemon`. Например так:

```
$ pkill ^dvtws$
$ /opt/zapret/nfq/dvtws --port=989 --daemon --dpi-desync=multisplit --dpi-desync-split-pos=2
```

Для перезапуска фаервола и демонов достаточно будет сделать:

```
$ /etc/rc.d/ipfw restart
```

tpws в прозрачном режиме

Краткая инструкция по запуску `tpws` в прозрачном режиме.

Note

Предполагается, что интерфейс LAN называется `em1`, WAN - `em0`.

Весь трафик

```
$ ipfw delete 100
$ ipfw add 100 fwd 127.0.0.1,988 tcp from me to any 80,443 proto ip4 xmit em0 not uid daemon
$ ipfw add 100 fwd ::1,988 tcp from me to any 80,443 proto ip6 xmit em0 not uid daemon
$ ipfw add 100 fwd 127.0.0.1,988 tcp from any to any 80,443 proto ip4 recv em1
$ ipfw add 100 fwd ::1,988 tcp from any to any 80,443 proto ip6 recv em1
$ /opt/zapret/tpws/tpws --port=988 --user=daemon --bind-addr=::1 --bind-addr=127.0.0.1
```

Трафик только на таблицу zapret, за исключением таблицы nozapret

```
$ ipfw delete 100
$ ipfw add 100 allow tcp from me to table\(nozapret\) 80,443
$ ipfw add 100 fwd 127.0.0.1,988 tcp from me to table\(zapret\) 80,443 proto ip4 xmit em0 not uid daemon
$ ipfw add 100 fwd ::1,988 tcp from me to table\(zapret\) 80,443 proto ip6 xmit em0 not uid daemon
```

```
$ ipfw add 100 allow tcp from any to table\(nozapret\) 80,443 recv em1
$ ipfw add 100 fwd 127.0.0.1,988 tcp from any to any 80,443 proto ip4 recv em1
$ ipfw add 100 fwd ::1,988 tcp from any to any 80,443 proto ip6 recv em1
$ /opt/zapret/tpws/tpws --port=988 --user=daemon --bind-addr=::1 --bind-addr=127.0.0.1
```

Note

Таблицы zapret, nozapret, ipban создаются скриптами из ipset по аналогии с Linux. Обновление скриптов можно забить в cron под root:

```
$ crontab -e
```

```
<...>
0 12 */2 * * /opt/zapret/ipset/get_config.sh
```

Caution

При использовании ipfw `tpws` не требует повышенных привилегий для реализации прозрачного режима. Однако, без рута невозможен bind на порты `< 1024` и смена UID/GID. Без смены UID будет рекурсия, поэтому правила ipfw нужно создавать с учетом UID, под которым работает `tpws`. Переадресация на порты `>= 1024` может создать угрозу перехвата трафика непривилегированным процессом, если вдруг `tpws` не запущен.

Запуск dvtws

Весь трафик

```
$ ipfw delete 100
$ ipfw add 100 divert 989 tcp from any to any 80,443 out not diverted xmit em0
# required for autottl mode only
$ ipfw add 100 divert 989 tcp from any 80,443 to any tcpflags syn,ack in not diverted recv em0
$ /opt/zapret/nfq/dvtws --port=989 --dpi-desync=multisplit --dpi-desync-split-pos=2
```

Трафик только на таблицу zapret, за исключением таблицы nozapret

```
$ ipfw delete 100
$ ipfw add 100 allow tcp from me to table\(nozapret\) 80,443
```

```
$ ipfw add 100 divert 989 tcp from any to table\(zapret\) 80,443 out not diverted not sockarg xmit em0
# required for autottl mode only
$ ipfw add 100 divert 989 tcp from table\(zapret\) 80,443 to any tcpflags syn,ack in not diverted not sockarg recv
$ /opt/zapret/nfq/dvtws --port=989 --dpi-desync=multisplit --dpi-desync-split-pos=2
```

PF в FreeBSD

Настройка аналогична **OpenBSD**, но есть важные нюансы.

- В **FreeBSD** поддержка PF в `tpws` отключена по умолчанию. Чтобы ее включить, нужно использовать параметр `--enable-pf`.
- Нельзя сделать ipv6 rdr на `::1`. Нужно делать на link-local адрес входящего интерфейса. Смотрите через `ifconfig` адрес `fe80:...` и добавляете в правило.
- Синтаксис `pf.conf` немного отличается. Более новая версия PF.
- Лимит на количество элементов таблиц задается так:

  ```
  $ sysctl net.pf.request_maxcount=2000000
  ```

- Сломан divert-to. Он работает, но не работает механизм предотвращения зацикливаний. Кто-то уже написал патч, но в `14-RELEASE` проблема все еще есть. Следовательно, на данный момент работа `dvtws` через PF невозможна.
  `/etc/pf.conf`:

  ```
  rdr pass on em1 inet6 proto tcp to port {80,443} -> fe80::31c:29ff:dee2:1c4d port 988
  rdr pass on em1 inet  proto tcp to port {80,443} -> 127.0.0.1 port 988
  ```

  ```
  $ /opt/zapret/tpws/tpws --port=988 --enable-pf --bind-addr=127.0.0.1 --bind-iface6=em1 --bind-linklocal=forc
  ```

Note

В PF не выходит делать rdr-to с той же системы, где работает proxy. Вариант с route-to не сохраняет мета информацию. Адрес назначения теряется. Поэтому этот вариант годится для squid, берущего адрес из протокола прикладного уровня, но не годится для tpws, полагающегося на метаданные ОС. Поддержка rdr-to реализована через `/dev/pf`, поэтому прозрачный режим **требует root**.

pfsense

Описание

pfsense основан на **FreeBSD** и использует фаервол PF, имеющий проблемы с divert. К счастью, модули ipfw и ipdivert присутствуют в поставке последних версий pfsense. Их можно подгрузить через `kldload`.

В некоторых более старых версиях pfsense требуется изменить порядок фаерволов через `sysctl`, сделав ipfw первым. В более новых эти параметры `sysctl` отсутствуют, но система работает как надо и без них. В некоторых случаях фаервол PF может ограничивать возможности `dvtws`, в частности в области фрагментации ip.

Присутствуют по умолчанию правила scrub для реассемблинга фрагментов.

Бинарики из `binaries/freebsd-x64` собраны под **FreeBSD 11**. Они должны работать и на последующих версиях **FreeBSD**, включая pfsense. Можно пользоваться `install_bin.sh`.

Автозапуск

Пример скрипта автозапуска лежит в `init.d/pfsense`. Его следует поместить в `/usr/local/etc/rc.d` и отредактировать на предмет правил ipfw и запуска демонов. Есть встроенный редактор `edit` как более приемлемая альтернатива `vi`.

Note

Поскольку `git` отсутствует, копировать файлы удобнее всего через `ssh`. `curl` присутствует по умолчанию. Можно скопировать zip с файлами zapret и распаковать в `/opt`, как это делается на других системах. Тогда `dvtws` нужно запускать как `/opt/zapret/nfq/dvtws`. Либо скопировать только `dvtws` в `/usr/local/sbin`. Как вам больше нравится.

Note

Скрипты ipset работают, крон есть. Можно сделать автообновление листов.

Note

Если вас напрягает бедность имеющегося репозитория, можно включить репозиторий от **FreeBSD**, который по умолчанию выключен.

Поменяйте `no` на `yes` в `/usr/local/etc/pkg/repos/FreeBSD.conf`

Можно установить весь привычный софт, включая `git`, чтобы напрямую скачивать zapret с github.

`/usr/local/etc/rc.d/zapret.sh` (chmod `755`):

```
#!/bin/sh

kldload ipfw
kldload ipdivert
```

```
# for older pfsense versions. newer do not have these sysctls
sysctl net.inet.ip.pfil.outbound=ipfw,pf
sysctl net.inet.ip.pfil.inbound=ipfw,pf
sysctl net.inet6.ip6.pfil.outbound=ipfw,pf
sysctl net.inet6.ip6.pfil.inbound=ipfw,pf

ipfw delete 100
ipfw add 100 divert 989 tcp from any to any 80,443 out not diverted xmit em0
pkill ^dvtws$
dvtws --daemon --port 989 --dpi-desync=multisplit --dpi-desync-split-pos=2

# required for newer pfsense versions (2.6.0 tested) to return ipfw to functional state
pfctl -d ; pfctl -e
```

Проблемы tpws

Что касается `tpws`, то видимо имеется некоторый конфликт двух фаерволов, и правила fwd в ipfw не работают. Работает перенаправление средствами PF как описано в разделе по **FreeBSD**. В PF можно изменять правила только целыми блоками - якорями (anchors). Нельзя просто так добавить или удалить что-то. Но чтобы какой-то anchor был обработан, на него должна быть ссылка из основного набора правил. Его трогать нельзя, иначе порушится весь фаервол. Поэтому придется править код скриптов pfsense.

1. Поправьте `/etc/inc/filter.inc` следующим образом:

```
<...>
/* MOD */
$natrules .= "# ZAPRET redirection\n";
$natrules .= "rdr-anchor \"zapret\"\n";

$natrules .= "# TFTP proxy\n";
$natrules .= "rdr-anchor \"tftp-proxy/*\"\n";
<...>
```

2. Напишите файл с содержимым anchor-а (например, `/etc/zapret.anchor`):

```
rdr pass on em1 inet  proto tcp to port {80,443} -> 127.0.0.1 port 988
rdr pass on em1 inet6 proto tcp to port {80,443} -> fe80::20c:29ff:5ae3:4821 port 988
```

`fe80::20c:29ff:5ae3:4821` замените на ваш link local адрес LAN интерфейса, либо уберите строчку, если ipv6 не нужен.

3. Добавьте в автозапуск `/usr/local/etc/rc.d/zapret.sh`:

```
$ pfctl -a zapret -f /etc/zapret.anchor
$ pkill ^tpws$
$ tpws --daemon --port=988 --enable-pf --bind-addr=127.0.0.1 --bind-iface6=em1 --bind-linklocal=force --split-pos
```

4. После перезагрузки проверьте, что правила создались:

```
$ pfctl -s nat
no nat proto carp all
nat-anchor "natearly/*" all
nat-anchor "natrules/*" all
<...>
no rdr proto carp all
rdr-anchor "zapret" all
rdr-anchor "tftp-proxy/*" all
rdr-anchor "miniupnpd" all

$ pfctl -s nat -a zapret
rdr pass on em1 inet proto tcp from any to any port = http -> 127.0.0.1 port 988
rdr pass on em1 inet proto tcp from any to any port = https -> 127.0.0.1 port 988
rdr pass on em1 inet6 proto tcp from any to any port = http -> fe80::20c:29ff:5ae3:4821 port 988
rdr pass on em1 inet6 proto tcp from any to any port = https -> fe80::20c:29ff:5ae3:4821 port 988
```

Note

Так же есть более элегантный способ запуска `tpws` через @reboot в cron и правило перенаправления в UI. Это позволит не редактировать код pfsense.

OpenBSD

tpws bind на ipv4

В `tpws` bind по умолчанию только на ipv6. Для bind на ipv4 нужно указать `--bind-addr=0.0.0.0`. Используйте `--bind-addr=0.0.0.0 --bind-addr=::` для достижения того же результата, как в других ОС по умолчанию. Но лучше все же так не делать, а сажать на определенные внутренние адреса или интерфейсы.

tpws для проходящего трафика (старая схема не работает в новых версиях)

В этом варианте tpws обращается явно к редиректору pf и пытается от него получить оригинальный адрес назначения. Как показывает практика, это не работает на новых версиях OpenBSD. Возвращается ошибка ioctl. Последняя проверенная версия, где это работает, - 6.8 . Между 6.8 и 7.4 разработчики сломали этот механизм.

`/etc/pf.conf`:

```
pass in quick on em1 inet  proto tcp to port {80,443} rdr-to 127.0.0.1 port 988
pass in quick on em1 inet6 proto tcp to port {80,443} rdr-to ::1 port 988
```

```
$ pfctl -f /etc/pf.conf
$ tpws --port=988 --user=daemon --bind-addr=::1 --bind-addr=127.0.0.1 --enable-pf
```

Note

В PF не выходит делать rdr-to с той же системы, где работает proxy. Вариант с route-to не сохраняет мета информацию. Адрес назначения теряется. Поэтому этот вариант годится для squid, берущего адрес из протокола прикладного уровня, но не годится для tpws, полагающегося на метаданные ОС. Поддержка rdr-to реализована через `/dev/pf`, поэтому прозрачный режим **требует root**.

tpws для проходящего трафика (новые системы)

В новых версиях предлагается использовать divert-to вместо rdr-to. Минимально проверенная версия, где это работает, 7.4. Может работать или не работать на более старых - исследование не проводилось.

`/etc/pf.conf`:

```
pass on em1 inet proto tcp to port {80,443} divert-to 127.0.0.1 port 989
pass on em1 inet6 proto tcp to port {80,443} divert-to ::1 port 989
```

tpws должен иметь бинд на точно такой адрес, который указан в правилах pf. `0.0.0.0` или `::` не работает.

```
$ pfctl -f /etc/pf.conf
$ tpws --port=988 --user=daemon --bind-addr=::1 --bind-addr=127.0.0.1
```

Note

Так же не понятно как делать divert с самой системы, где работает tpws.

Запуск dvtws

Весь трафик

`/etc/pf.conf`:

```
pass in  quick on em0 proto tcp from port {80,443} flags SA/SA divert-packet port 989 no state
pass in  quick on em0 proto tcp from port {80,443} no state
pass out quick on em0 proto tcp to   port {80,443} divert-packet port 989 no state
```

```
$ pfctl -f /etc/pf.conf
$ ./dvtws --port=989 --dpi-desync=multisplit --dpi-desync-split-pos=2
```

Трафик только на таблицу zapret, за исключением таблицы nozapret

`/etc/pf.conf`:

```
set limit table-entries 2000000
table <zapret> file "/opt/zapret/ipset/zapret-ip.txt"
table <zapret-user> file "/opt/zapret/ipset/zapret-ip-user.txt"
table <nozapret> file "/opt/zapret/ipset/zapret-ip-exclude.txt"
pass out quick on em0 inet  proto tcp to   <nozapret> port {80,443}
pass in  quick on em0 inet  proto tcp from <zapret>  port {80,443} flags SA/SA divert-packet port 989 no state
pass in  quick on em0 inet  proto tcp from <zapret>  port {80,443} no state
pass out quick on em0 inet  proto tcp to   <zapret>  port {80,443} divert-packet port 989 no state
pass in  quick on em0 inet  proto tcp from <zapret-user>  port {80,443} flags SA/SA divert-packet port 989 no state
pass in  quick on em0 inet  proto tcp from <zapret-user>  port {80,443} no state
pass out quick on em0 inet  proto tcp to   <zapret-user>  port {80,443} divert-packet port 989 no state
table <zapret6> file "/opt/zapret/ipset/zapret-ip6.txt"
table <zapret6-user> file "/opt/zapret/ipset/zapret-ip-user6.txt"
table <nozapret6> file "/opt/zapret/ipset/zapret-ip-exclude6.txt"
pass out quick on em0 inet6 proto tcp to   <nozapret6> port {80,443}
```

```
pass in  quick on em0 inet6 proto tcp from <zapret6> port {80,443} flags SA/SA divert-packet port 989 no state
pass in  quick on em0 inet6 proto tcp from <zapret6> port {80,443} no state
pass out quick on em0 inet6 proto tcp to   <zapret6> port {80,443} divert-packet port 989 no state
pass in  quick on em0 inet6 proto tcp from <zapret6-user>  port {80,443} flags SA/SA divert-packet port 989 no state
pass in  quick on em0 inet6 proto tcp from <zapret6-user>  port {80,443} no state
pass out quick on em0 inet6 proto tcp to   <zapret6-user> port {80,443} divert-packet port 989 no state
```

```
$ pfctl -f /etc/pf.conf
$ ./dvtws --port=989 --dpi-desync=multisplit --dpi-desync-split-pos=2
```

Проблемы с badsum

OpenBSD принудительно пересчитывает tcp checksum после divert, поэтому скорее всего `dpi-desync-fooling=badsum` у вас не заработает. При использовании этого параметра `dvtws` предупредит о возможной проблеме.

Особенность отправки fake пакетов

В **OpenBSD** `dvtws` все фейки отсылает через divert socket, поскольку эта возможность через raw sockets заблокирована. Видимо PF автоматически предотвращает повторный заворот diverted фреймов, поэтому проблемы зацикливания нет.

divert-packet автоматически вносит обратное правило для перенаправления. Трюк с no state и in правилом позволяет обойти эту проблему, чтобы напрасно не гнать массивный трафик через `dvtws`.

Перезагрузка PF таблиц

Скрипты из ipset не перезагружают таблицы в PF по умолчанию.

Чтобы они это делали, добавьте параметр в `/opt/zapret/config`:

```
LISTS_RELOAD="pfctl -f /etc/pf.conf"
```

Более новые версии `pfctl` понимают команду перезагрузить только таблицы. Но это не относится к **OpenBSD**. В новых **FreeBSD** есть.

```
$ pfctl -Tl -f /etc/pf.conf
```

Important

Не забудьте выключить сжатие gzip: `GZIP_LISTS=0`

Important

Если в вашей конфигурации какого-то файла листа нет, то его необходимо исключить из правил PF. Если вдруг листа нет, и он задан в pf.conf, будет ошибка перезагрузки фаервола.

Note

После настройки обновление листов можно поместить в cron:

```
$ crontab -e
```

```
<...>
0 12 */2 * * /opt/zapret/ipset/get_config.sh
```

https://forum.lissyara.su/freebsd-f8/obhod-dpi-na-shlyuze-t46842.html

---

Revision #2
Created 14 October 2025 09:57:32 by buzz
Updated 14 October 2025 12:14:23 by buzz