

Настройка

- [Настройка Redis](#)
- [Настройка компонентов ISP Go](#)
- [Настройка nginx](#)
- [Настройка автоматического обновления базы доменов](#)
- [Настройка отправки статистики](#)
- [Настройка страницы блокировки](#)

Настройка Redis

В файле `/etc/redis/redis.conf` необходимо указать максимальный объем памяти, используемой для хранения данных, и алгоритм работы при достижении лимита. Во избежание потери данных, следует использовать такие значения:

```
maxmemory 2GB
```

```
maxmemory-policy noeviction
```

Для применения настроек следует перезапустить Redis:

```
service redis-server restart
```

Настройка компонентов ISP Go

Все компоненты ISP Go читают один файл настроек: `/etc/isp-go/config.ini`. Он состоит из нескольких секций.

Секция [dnscache]

Секция используется `isp-go-dnsproxy` и содержит единственный ключ `forward`. В качестве значения следует указать IP-адрес и порт кеширующего DNS-сервера, на который следует перенаправлять все незаблокированные DNS-запросы.

Пример:

```
[ dnscache]

forward = 127. 0. 0. 1: 53
```

Секция [proxy]

Секция используется `isp-go-dnsproxy` и содержит ключи `listen`, `blockpage-ip`, `log` и `pid`.

Ключ `listen` содержит IP-адрес и порт, на котором `isp-go-dnsproxy` должен принимать запросы от клиентов. Он может быть продублирован чтобы принимать запросы сразу на нескольких адресах, например, IPv4 и IPv6 одновременно. Число ключей, равно как и прослушиваемых адресов, неограниченно.

Ключ `blockpage-ip` содержит IP-адрес, который следует отдавать клиентам в ответах на заблокированные запросы. Иными словами, это IP-адрес, на котором `nginx` принимает запросы к странице блокировки. В типичном случае, когда фильтрующий DNS-сервер и `nginx` запущены на одном сервере, следует и там и там прописать IP-адрес одного из сетевых интерфейсов сервера. Ключ можно продублировать, чтобы указать альтернативный адрес страниц блокировок, например, для IPv6. Допускается использование не менее одного ключа только для перенаправлений A или только AAAA пакетов, но не более двух для поддержки обеих версий IP.

Ключи `log` и `pid` содержат абсолютные пути до `log`-файла и `pid`-файла, соответственно. Чтобы не потерять совместимость с `init`-скриптами, входящими в состав пакета, путь до `pid`-файла изменять запрещено.

Пример:

```
[ proxy]
```

```
listen = 192.168.5.1:53 ; IPv4
```

```
listen = [ 4321: a: bcde: 1:: 2020]:53 ; IPv6
```

```
blockpage-ip = 192.168.5.1 ; IPv4 для перенаправления A
```

```
blockpage-ip = abcd:1234:zyxw:9876 ; IPv6 для перенаправления AAAA
```

```
log = /var/log/isp-go/isp-go-dnsproxy.log
```

```
pid = /var/log/isp-go/isp-go-dnsproxy.pid
```

Секция [datafiles]

Секция содержит ключи path, file и cats, при этом ключ file может встречаться несколько раз.

Ключ path содержит абсолютный путь к каталогу, в котором находится база доменов SkyDNS. Файлы, составляющие базу доменов, перечислены по одному в ключах file. Порядок ключей важен - каждый файл рассматривается как набор поправок к файлам, указанным после него. Для сохранения работоспособности базы, в ключах path и file запрещено указывать значения, отличные от таковых по умолчанию.

Ключ cats содержит абсолютный путь до директории со списками категорий.

Редактирование файла `catgroups.json`, входящего в состав пакета, не допускается, т.к. изменения пропадут при обновлении пакета ISP Go. В случае необходимости сокращения части категорий или перевода названий категорий на другой язык, необходимо создать копию файла `catgroups.json` и внести изменения в нее. При этом номера категорий (50: ...) изменять нельзя, т.к. они должны соответствовать содержимому базы доменов SkyDNS.

Пример:

```
[ datafiles]
```

```
path = /var/lib/isp-go/filter/
```

```
file = host2cat-fast.dat
```

```
file = host2cat.dat
```

```
cats = /usr/share/isp-go/config/
```

По умолчанию категории выводятся на русском языке. Для получения категорий на английском языке необходимо указать файл `catgroups_en.json` (входит в состав пакета).

Секция [blockpage]

Секция используется веб-приложением **страничка блокировки** и содержит ключи listen templates, log и pid.

Ключ `listen` содержит IP-адрес (обычно 127.0.0.1) и порт, на котором демон `isp-go-blockpage` принимает HTTP-запросы, предназначенные для страницы блокировки. Важно: демон `isp-go-blockpage` не принимает запросы от пользователей напрямую, этим занимается Nginx.

Ключ `templates` содержит абсолютный путь до каталога с шаблонами страницы блокировки. Изменять непосредственно шаблоны, поставляемые в составе пакета, нельзя (так как изменения будут потеряны при обновлении ISP Go), но можно скопировать весь каталог `/usr/share/isp-go/templates` под другим именем и изменять файлы в копии.

Ключи `log` и `pid` содержат абсолютные пути до log-файла и pid-файла, соответственно. Чтобы не потерять совместимость с init-скриптами, входящими в состав пакета, путь до pid-файла изменять запрещено.

Пример:

```
[blockpage]

listen = 127.0.0.1:8081

templates = /usr/share/isp-go/templates/

log = /var/log/isp-go/isp-go-blockpage.log

pid = /var/run/isp-go/isp-go-blockpage.pid
```

Секция [api]

Секция используется веб-приложением **ISP Go API** и содержит ключи `listen`, `log` и `pid`.

Ключ `listen` содержит IP-адрес (обычно 127.0.0.1) и порт, на котором демон `isp-go-api` принимает HTTP-запросы, предназначенные для страницы блокировки. Адрес или порт должен отличаться от значения, используемого в секции `[blockpage]`.

Важно: прослушивание на доступном извне IP-адресе будет проблемой в безопасности. Демон `isp-go-api` не содержит никаких механизмов авторизации, поэтому любой, кто может отправить запрос, может внести любые изменения в пользовательские настройки (включая чужие). Для предотвращения такой ситуации рекомендуется использовать здесь IP-адрес 127.0.0.1, а предоставление доступа извне (с авторизацией) реализовать на уровне Nginx.

Ключи `log` и `pid` содержат абсолютные пути до log-файла и pid-файла, соответственно. Чтобы не потерять совместимость с init-скриптами, входящими в состав пакета, путь до pid-файла изменять запрещено.

Пример:

```
[api]

listen = 127.0.0.1:8080
```

```
log = /var/log/isp-go/isp-go-api.log
```

```
pid = /var/run/isp-go/isp-go-api.pid
```

Секция [common]

Секция используется всеми тремя демонами и содержит ключи `redis-ip` и `redis-port`.

Ключи `redis-ip` и `redis-port` задают, к какому Redis-серверу должны подключаться демоны, входящие в состав ISP Go. Исходя из соображений производительности, рекомендуется запускать redis-сервер на той же машине, где установлен ISP Go.

Пример:

```
[common]
```

```
redis-ip = 127.0.0.1
```

```
redis-port = 6379
```

Применение настроек

После изменения конфигурационного файла на работающем сервере необходимо перезапустить службы, входящие в состав ISP Go:

```
service isp-go-dnsproxy restart  
service isp-go-blockpage restart  
service isp-go-api restart
```

Настройка nginx

Роль Nginx в решении для провайдеров состоит в следующем:

- отделение запросов к API от запросов к странице блокировки
- проксирование этих запросов на соответствующие веб-приложения
- ограничение доступа к API

Для корректного разделения запросов необходимо прописать доменное имя, используемое для управления фильтрацией через API, в директиву `server_name` в файле `/etc/nginx/sites-available/isp-go-api`, вместо значения `api.ispgo`, которое там прописано по умолчанию. Все остальные запросы пойдут на виртуальный хост, настроенный в файле `/etc/nginx/sites-available/isp-go-blocked`, за счет наличия там модификатора `default_server` в директиве `listen`. Допускается создавать другие виртуальные хосты, со следующими оговорками:

- в каждом виртуальном хосте должно быть прописано имя сервера с помощью директивы `server_name`;
- не допускается использование модификатора `default_server`;
- не рекомендуется использование протокола `https` - в этом случае, вместо сброса соединения, у пользователей, обращающихся к запрещенным сайтам по `https`, будет появляться предупреждение о невалидном сертификате, и неизбежно возникнут (необоснованные) обвинения от пользователей в перехвате защищенного трафика.

Для корректного проксирования необходимо, чтобы адреса и порты, упоминаемые в директиве `proxy_pass` в файлах `/etc/nginx/sites-available/isp-go-api` и `/etc/nginx/sites-available/isp-go-blocked`, соответствовали адресам и портам, которые прослушиваются соответствующими веб-приложениями. По умолчанию это так. См. ключи `listen` в секциях `[blockpage]` и `[api]` в файле `/etc/isp-go/config.ini`.

Доступ к API ограничивается с помощью директив `allow` и `deny`. Директивы обрабатываются по очереди сверху вниз до первого соответствия. Конфигурация по умолчанию разрешает доступ только с адреса `127.0.0.1`. Следует разрешить доступ с сервера, на котором установлена биллинговая система. Ни в коем случае нельзя разрешать доступ к API с недоверенных (в том числе пользовательских) систем, т.к. при наличии доступа к API злоумышленник может изменять любые настройки фильтрации у любых пользователей.

Для применения настроек надо заставить Nginx перечитать конфигурационные файлы:

```
service nginx reload
```

Более подробную информацию о настройке nginx можно найти на странице проекта:

[*http://nginx.org/ru/docs/*](http://nginx.org/ru/docs/)

Настройка автоматического обновления базы доменов

В ходе первичной установки пакета в каталог `/var/lib/isp-go/filter/` копируется демонстрационная версия базы доменов. Для полноценного использования ISP Go необходимо заменить ее на настоящую и настроить автообновление.

Обновление базы доменов осуществляется по cron'у с использованием rsync. Для авторизации доступа к серверу `skydns.ru` используется ssh-ключ. Чтобы автообновление заработало, необходимо:

1. Сгенерировать ssh-ключ, который будет использоваться для скачивания обновлений базы доменов:

```
mkdir skydns-key
cd skydns-key
ssh-keygen -t rsa -N "" -f id_rsa
```

В результате получатся файлы `id_rsa` (закрытый ключ, который надо держать в строгом секрете и не терять) и `id_rsa.pub` (открытый ключ).

2. Выслать получившийся файл `id_rsa.pub` по электронной почте. `id_rsa` высылать нам не надо.
3. Дождаться, пока будет произведена авторизация SSH-ключа на сервере SkyDNS.
4. Скопировать файлы `id_rsa` и `id_rsa.pub` в каталог, где их ищет скрипт обновления:

```
mkdir -p -m 0755 /var/lib/isp-go/.ssh
cd skydns-key
cp id_rsa id_rsa.pub /var/lib/isp-go/.ssh/
chown -R isp-go:isp-go /var/lib/isp-go/.ssh
```

5. Подождать 1 час. Убедиться, что в каталоге `/var/lib/isp-go/filter` обновились файлы `host2cat.dat` и `host2cat-fast.dat`. Обновить dat-файлы можно также вручную, для этого достаточно выполнить команды:

```
su isp-go -c 'rsync -rtv --progress skydns-isp@skydns.ru: host2cat.dat ~/filter/'
su isp-go -c 'rsync -rtv --progress skydns-isp@skydns.ru: host2cat-fast.dat ~/filter/'
```

Настройка отправки статистики

Для корректного обновления базы доменов требуется настройка отправки статистики isp-go.

Для этого необходимо разрешить устанавливать исходящие соединения с сервера isp-go на www.skydns.ru по tcp-порту 443 и выполнить все шаги из п Настройка автоматического обновления базы доменов.

Для проверки корректности отправки статистики на сервере isp-go нужно выполнить

```
curl -f -X POST --data "key=`cut -d ' ' -f 2 /var/lib/isp-go/.ssh/id_rsa.pub | base64 -d | md5su
```

Корректным ответом будет ответ ok.

Настройка страницы блокировки

По умолчанию ISP Go поставляется с минимальным, строгим и аскетичным оформлением страницы блокировки. Чтобы изменить это оформление, надо отредактировать HTML-шаблоны, которые находятся в `/usr/share/isp-go/templates/`. Где `base.html` - основной файл шаблона, а остальные от него наследуются. Синтаксис шаблонов описан в руководстве по языку Go: <https://golang.org/pkg/text/template/>, <https://golang.org/pkg/html/template/>. Доступны следующие переменные:

- `Domain`: домен, упомянутый в заголовке `Host`: запроса, т.е. тот, к которому обратился браузер.
- `Cats`: массив с именами категорий, из-за которых сайт заблокирован. Переменная доступна только в шаблоне `blocked_by_category.html`.

В случае необходимости вставки картинок, они должны быть выложены на отдельный виртуальный хост, и в теге должен указываться их полный URL, включая имя этого картиночного хоста.

Изменения вступают в силу после перезапуска `isp-go-blockpage` (`service isp-go-blockpage restart`)