4. Configuration

Redis configuration

Set the maximum size of memory allocated for data storage and policy of work when this limit will be achieved in `/etc/redis/redis.conf` the file. To prevent loss of data you should use these settings:

```
maxmemory 2GB
maxmemory-policy noeviction
```

To apply the settings restart **Redis server** service:

```
service redis-server restart
```

ISP Go modules configuration

All the modules read the same configuration file `/etc/isp-go/config.ini`
The file consists of several sections.

[dnscache] section

[dnscache] section

This section is used by `isp-go-dnsproxy` and contains only one key - forward. As a value, you should set an IP address and a port of caching DNS server which will be used to resolve all non-blocked DNS requests.

Example:

```
[dnscache]
forward = 8.8.8.8:53
```

[proxy] section

[proxy] section

This section is used by `isp-go-dnsproxy` and contains listen ***blockpage-ip***, ***log***, and ***PID*** keys. The 'Listen' key contains an IP address and a port on which `isp-go-dnsproxy` should get DNS requests from users. It can be duplicated to accept requests on several addresses at once, for example, IPv4 and IPv6 at the same time. The number of keys, as well as the number of listening addresses, is unlimited

`blockpage-ip` key contains an IP address of a block page, which should be passed to users in answers to blocked requests. This is the IP address where **Nginx** accepts requests to the block page. In general, case when filtering DNS server and **Nginx** are on the same server you should set for this key the same IP address as for the listening key. The key can be duplicated to specify an alternative address for the block pages, for example, for IPv6. It is allowed to use at least one key only for A redirects or only AAAA packets, but no more than two to support both IP versions.

> `log, pid` keys contain the absolute path to the log file and PID file accordingly. To prevent loss of compatibility with init scripts from the ISP Go package you should not change the path to the PID file.

Example:

```
[proxy]
listen = 192.168.5.1:53 ; IPv4
listen = [4321:a:bcde:1::2020]:53 ; IPv6
blockpage-ip = 192.168.5.1 ; IPv4 to forwarding A
blockpage-ip = abcd:1234:zyxw::9876 ; IPv6 to forwarding AAAA
log = /var/log/isp-go/isp-go-dnsproxy.log
pid = /var/log/isp-go/isp-go-dnsproxy.pid
```

[datafiles] section

[datafiles] section

This section contains '**path', 'file', 'cats**' keys. **'file'** key can be used several times in this section.

The **'path'** key contains the absolute path to the directory with the SafeDNS domain database. Files, that database is made of, should be enumerated in the **'file'** keys. The order of these keys is important for correct work. Each top-listed file is processed as a correction to all bottom-listed files. To maintain the correct work of the database we prohibit the change of default values for these keys.

The **‘cats’** key contains the absolute path to the JSON file with the list of supported categories. You cannot change this file, because all changes made to it, will be lost on the ISP Go package update. If you need to hide some categories or translate them to some other language you should create a copy of the **catgroups.json** file and make all changes in the copy. The numbers of the categories should not be changed because they are linked to the content of the master database of SafeDNS.

Example:

```
[datafiles]
path = /var/lib/isp-go/filter/
file = host2cat-fast.dat file = host2cat.dat
cats = /usr/share/isp-go/config/
```

[blockpage] section

[blockpage] section

This section is used by the `isp-go-blockpage` application and contains '**listen**', '**templates**', '**log**' , and '**pid**' keys.
‘listen’ key contains an IP address (usually 127.0.0.1) and a port that `isp-go-blockpage` daemon listens on for HTTP requests to block page.

> Daemon isp-go-blockpage does not accept requests from users. All requests should be passed through **Nginx**.

The **‘templates’** key contains the absolute path to the directory with templates of block page. Do not change templates installed with the package, because all changes made will be lost on the ISP Go package update. We recommend copying the whole directory `/usr/share/isp-go/templates` and changing templates in this copy.

‘log’ and **‘pid’** keys contain the absolute path to log and PID files accordingly. To prevent loss of compatibility with init scripts from the ISP Go package you should not change the path to pidfile.

Example:

```
[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
```

[api] section

[api] section

This section is used by the web application `isp-go-api` and contains '**listen**', '**log**', and '**pid**' keys.
‘listen’ key contains an IP address (usually 127.0.0.1) and a port that `isp-go-api` daemon listens on for HTTP requests to API. An IP address or a port should be different from the set in the **[blockpage]** section

> Listening on an externally accessible IP address will be a security problem. The isp-go-api daemon does not contain any authorization mechanisms, so anyone who can send a request can make any changes to user settings (including someone else's). To prevent such a situation, it is recommended to use the IP address 127.0.0.1 here and to implement external access (with authorization) at the Nginx level.

Example:

```
[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
```

[common] section

[common] section

The section is used by all three daemons and contains the keys '**redis-ip**' and '**redis-port**'.

The '**redis-ip**' and '**redis-port**' keys specify which **Redis** server the daemons that are part of ISP Go should connect to. For performance reasons, it is recommended to run the **Redis** server on the same machine where ISP Go is installed.

Example:

```
[common]
redis-ip = 127.0.0.1
redis-port = 6379
```

Enabling ISP Go services

To enable ISP Go services automatic startup please run the following commands:

```
systemctl enable isp-go-dnsproxy
systemctl enable isp-go-blockpage
systemctl enable isp-go-api
```

Applying settings

After changing the configuration file you should restart all ISP Go services:

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

Nginx configuration

Nginx is used in ISP Go for the following tasks:
• separation of API requests from requests to the block page and to the administrative web interface
• proxying requests to corresponding web applications
• restricting access to the API and to the administrative web interface

For correct separation of requests, you need to register the domain name used to manage filtering through the API in the `server_name` directive of the file `/etc/nginx/sites-available/isp-go-api` instead of the value 'api.ispgo'.

All other requests will go to the virtual host configured in the file `/etc/nginx/sites-available/isp-go-blocked` due to the presence of the `default_server` modifier in the `listen` directive.

You can create additional virtual hosts with the following exceptions:
• each additional virtual host should contain the `server_name` directive
• `default_server` option cannot be used

• HTTPS usage is not recommended, because users who are requesting blocked websites via HTTPS will get browser warnings of an invalid SSL certificate.

IP addresses and ports in the `proxy_pass` directive in **Nginx** configuration files should correspond with IP addresses and ports on which web applications were launched (see **listen** key in `/etc/isp-go/config.ini` file).

API access is restricted using the `allow` and `deny` directives. Directives are processed in turn from top to bottom until the first match. The default configuration allows access only from the address **127.0.0.1**. You must allow access from the server where the billing system is installed.

> In no case, access should be allowed to the API from untrusted (including user) systems, because if access to the API is provided an attacker can change any filtering settings for any users.

To apply changes reload **Nginx**:

```
service nginx reload
```

Database update

In the installation process demo version of the database will be copied to `/var/lib/isp-go/filter/` folder. For production deployment, you should replace the demo version with the full one and configure automatic updates.
The domain database is updated by **cron** using **rsync**. You need an **ssh key** to authorize access to the **safedns.com** server. To get auto-update to perform the following steps:

Generate an ssh key that will be used to download domain database updates:

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

As a result, the files `id_rsa` (private key, which must be kept strictly secret and not lost) and `id_rsa.pub` (public key) will be created.

Send the created `id_rsa.pub` file to the technical support team at support@safedns.com Don't need to send `id_rsa` antwhere!

The technical support team will notify you when the ssh key will be authorized on the SafeDNS server.

Copy the `id_rsa` and `id_rsa.pub` files to the directory where the update script is looking for them:

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

Wait for 1 hour and make sure that `host2cat.dat` and `host2cat-fast.dat` files in the directory `/var/lib/isp-go/filter` have been updated.

You can also update the data-files manually, just run the following commands:

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

> The server should have access to www.safedns.com on TCP port 443.

Setting up sending the statistics

To do this, you need to allow outgoing connections from the isp-go server to www.safedns.com on **TCP** port **443** and perform all the steps from Setting up automatic updating of the domain database (previous part).

To check the correctness of sending statistics on the isp-go server, please perform:

```
curl -f -X POST --data "key=`cut -d ' ' -f 2 /var/lib/isp-go/.ssh/id_rsa.pub | base64 -d |
md5sum | cut -d ' ' -f1`&count=`redis-cli --raw hlen ip`" https://www.safedns.com/isp-kit-dog/
```

OK is the correct answer to this command.

Block page design

By default, ISP Go comes with a minimal, strict, and ascetic design of the lock page. To change this design, you need to edit the **HTML** templates that are located at `/usr/share/isp-go/templates/`, where `base.html` – is the main template file, and the rest are inherited from it. The syntax of templates is described in the Go language guide:

- https://golang.org/pkg/text/template/
- https://golang.org/pkg/html/template/

The following variables are available:

- Domain: requested hostname in the Host field of HTTP request header
- Cats: array with category names of the blocked websites. This variable is allowed to use with the `blocked_by_category.html`

template file only.

To add images to a block page we recommend setting a separate virtual host for image storing and using an absolute **URL** to an image with `<img>` tag (example: `<img>http://img.isp.com/block-img.png</img>`)

To apply changes restart the `isp-go-blockpage` service:

```
service isp-go-blockpage restart
```

Peculiarities of listening on specific IP addresses

It is possible to use the **0.0.0.0** IP address in the listen key of a `[proxy]` section of the configuration file if you use caching DNS server installed on another server. In this case, `isp-go-dnsproxy` will process requests on all network interfaces.
If you have caching DNS server installed on the same server (and listening on **127.0.0.1** IP address), you should designate a specific IP address from one of the network interfaces.

User activity statistics

Statistics are recorded to a **CSV** file and imported into the **PostgreSQL** database every 5 minutes. Statistics are stored in the database in "raw" and aggregated form. "Raw" statistics are stored for 1 day. Aggregated statistics are stored for 3 months.
The statistics are recalculated every five minutes.

If a user has more than one IP address, then statistics for all addresses are summarized. Likewise for anonymous users.
The following reports are available:
• number of requests by hours and days for the period;
• number of requests to the top 100 domains;
• detailed statistics by domains and days;
• "Raw" statistics for the last hour.
Statistics are accessed through requests to the **ISP Go REST API.**

Created 2022-09-07 17:57:34 UTC
Updated 2023-10-01 01:04:55 UTC by Mark Woodman