Linguard

Release 1.0.0

José Antonio Mazón San Bartolomé

Oct 21, 2021

CONTENTS

| 1 | Key features 3 | | | | | |
|---|-----------------------|----|--|--|--|--|
| 2 | Contents | 5 | | | | |
| | 2.1 Installation | 5 | | | | |
| | 2.2 How does it look? | 5 | | | | |
| | 2.3 In depth | 12 | | | | |
| | 2.4 Contributing | 17 | | | | |
| | 2.5 Changelog | 19 | | | | |
| 3 | Indices and tables | 23 | | | | |

Linguard aims to provide a clean, simple yet powerful web GUI to manage your WireGuard server, and it's powered by Flask.

CHAPTER

ONE

KEY FEATURES

- Management of Wireguard interfaces and peers via web. Interfaces can be created, removed, edited, exported and brought up and down directly from the web GUI. Peers can be created, removed, edited and downloaded at anytime as well.
- Display stored and real time traffic data using charts (storage of traffic data may be manually disabled).
- Display general network information.
- Encrypted user credentials (AES).
- Easy management through the linguard systemd service.

CHAPTER

TWO

CONTENTS

2.1 Installation

2.1.1 As a systemd service

- 1. Download any release.
- 2. Extract it and run the installation script:

```
chmod +x install.sh
sudo ./install.sh
```

3. Run Linguard:

```
sudo systemctl start linguard.service
```

2.1.2 Using docker

- 1. Download the docker-compose.yaml file from the repository.
- 2. Run Linguard:

sudo docker-compose up -d

Note: You can check all available tags here.

2.2 How does it look?

Here are a bunch of screenshots:





| amaranth-mink | | home | | iPhone | | | | raspberry pi | | | |
|--------------------------------------|---------------|-------------|------------------|--------------------------------------|------|-------------|-------|--------------|----------|------------|------------------|
| Information provided by wg at 23:16. | | | | Information provided by Mg at 23:16. | | | | | | | |
| m Interfaces | | | | | | | | | | | + |
| Show 10 + entries | | | | | | | | | Search: | | |
| Name | 1 Description | | IPv4 | | List | ten port | Peers | | Status 斗 | Actions | |
| amaranth-mink | None | | 172.18.213.213/2 | 3 | 573 | 330 | 0 | | down | ± / 🗉 | |
| home | None | | 10.0.0.1/24 | | 561 | 107 | 2 | | up | ± / = | |
| Showing 1 to 2 of 2 entries | | | | | | | | Previous 1 | Next | | |
| Information provided at 23:16. | | | | | | | | | | | |
| Le Peers | | | | | | | | | | | + |
| Show 10 ¢ entries | | | | | | Search | | | Search: | | |
| Name 👫 Ir | terface th | Description | | IPv4 | | Primary DNS | | Secondary DN | IS th | Actions | |
| iPhone h | ome | None | | 10.0.3/24 | | 8.8.8.8 | | None | | ± / = | |
| raspberry pi h | ome | None | | 10.0.0.2/24 | | 8.8.8.8 | | None | | ± / = | |
| Showing 1 to 2 of 2 entries | | | | | | | | | | Previous 1 | Next |
| Information provided at 23:16. | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | Github · License |

Fig. 2: Dashboard (2)

| = | | | | | |
|----------|-------------------|-------------|-----------------------------|--------|----------------------------------|
| Netwo | rk | | | | |
| nterface | ŝ | | | | |
| lo | | enp0s31f6 | | docker | |
| Status | up . | Status | up | Status | down |
| IPv4 | 127.0.0.1/8 | IPv4 | 192.168.0.47/24 | IPv4 | 172.17.0.1/16 |
| IPv6 | ::1/128 | IPv6 | fe80::e54f:a25:e7ce:a99d/64 | IPv6 | None |
| MAC | 00:00:00:00:00:00 | MAC | 4c:cc:6a:bd:23:50 | MAC | 02:42:68:d5:e2:9f |
| Flags | LOOPBACK | Flags | BROADCAST, MULTICAST | Flags | NO-CARRIER, BROADCAST, MULTICAST |
| | | | | | |
| home | | amaranth-n | nink | | |
| Status | up | Status | down | | |
| IPv4 | 10.0.0.1/24 | IPv4 | 172.18.213.213/28 | | |
| | | | | | |
| IPv6 | None | IPv6 | None | | |
| IPv6 | None | IPv6 MAC | None | | |

Fig. 3: Network interfaces

| | Status | up | Status down | | | | | | | |
|----------------|---|--------------------|-------------|--|-----------|----------------|----------|-------|--------|------------------|
| | IPv4 | 10.0.0.1/24 | | | IPv4 | 172.18.213.213 | 8/28 | | | |
| | IPv6 | None | | | IPv6 | None | None | | | |
| | MAC | None | | | MAC | None | | | | |
| | Flags | POINTOPOINT, NOARF | • | | Flags | None | None | | | |
| | | | | | | | | | | |
| | More | | | | More | | | | | |
| | | | | | | | | | | |
| | Information provided by ±p and wg at 22:17. | | | | | | | | | |
| | Routing table | | | | | | | | | |
| | | | | | | | | | | |
| | Destination | | Gateway | | Device | | Protocol | Scope | Metric | Flags |
| | default | | 192.168.0.1 | | enp0s31f6 | | dhcp | None | 100 | None |
| | 10.0.0/24 | | default | | home | | kernel | link | None | None |
| | 169.254.0.0/16 | | default | | enp0s31f6 | | None | link | 1000 | None |
| | 172.17.0.0/16 | | default | | docker0 | | kernel | link | None | linkdown |
| | 192.168.0.0/24 | | default | | enp0s31f6 | | kernel | link | 100 | None |
| | | | | | | | | | | |
| Welcome, admin | Information provided by i | at 23:17. | | | | | | | | |
| Linguard 0.2.0 | | | | | | | | | | Othub - Lissan |
| Linguard 0.2.0 | | | | | | | | | | Github - License |



| nguard | Ξ | | | | | | | | | | ± - |
|---------------------|---------------|---------------------------------|-------------------|-------------|-------------|--------|----------|-------------|---------------|---------|------------|
| | Wi | reguard | | | | | | | | | |
| | • | iterfaces | | | | | | | | | + |
| | amaranth-mink | | home | | | | | | | | |
| | | Status | down | | Status | | up | | | | |
| Wireguard | | Description | None | | Description | ı | None | | | | |
| | | IPv4 | 172.18.213.213/28 | | IPv4 | | 10.0.0.1 | 1/24 | | | |
| | _ | Listen port | 57330 | | Listen port | | 56107 | | | | |
| ¹ Themes | | Peers | 0 | | Peers | | 2 | | | | |
| | | Start Restart | | | Stop | estart | | | | | |
| | | | | ± / 1 | auto | | | | | | |
| | Sta | art all Stop all | | | | | | | | | |
| | Inform | nation provided by wg at 23:17. | | | | | | | | | |
| | 41 (| Peers | | | | | | | | | + |
| | Show | v 10 ¢ entries | | | | | | | Search | | |
| | Na | me 1. | Interface | Description | | IPv4 | | Primary DNS | Secondary DNS | Actions | |

Fig. 5: Wireguard interfaces

| at Network | Status | down | | Status | up | | | | | |
|--------------------------------------|--------------------------------|-------------------|-------------|------------|-----------|-------------|--|---------------|----------|------------------|
| Wireguard | Description | None | | Descriptio | n Non | e | | | | |
| CUSTOMIZATION | IPv4 | 172.18.213.213/28 | | IPv4 | 10.0 | .0.0.1/24 | | | | |
| Settings | Listen port | 57330 | | Listen por | t 5610 | 56107 | | | | |
| 🖾 Themes | Peers | 0 | | Peers | 2 | | | | | |
| | Start Restart | | | Stop R | estart | | | | | |
| | | | ± / = | auto | | ± / = | | | | |
| Start all Stop all | | | | | | | | | | |
| Information provided by wg at 23:17. | | | | | | | | | | |
| | Le Peers | | | | | | | | | + |
| | Show 10 ¢ entries | | | | | | | Search: | | |
| | Name 1 | Interface | Description | | IPv4 | Primary DNS | | Secondary DNS | Actions | |
| | iPhone | home | None | | 10.0.3/24 | 8.8.8.8 | | None | ± / = | |
| | raspberry pi | home | None | | 10.0.2/24 | 8.8.8.8 | | None | ± / = | |
| | Showing 1 to 2 of 2 entries | | | | | | | | Previous | Next |
| Welcome, admin | Information provided at 23:17. | | | | | | | | | |
| Linguard 0.2.0 | | | | | | | | | | Github - License |

Fig. 6: Wireguard peers

| Linguard | = | | <u>*</u> * |
|------------------------|--|---|------------|
| summary Dashboard | Interface | | ир |
| About | X Actions | | |
| SYSTEM | Stop | | |
| 응 Network Wireguard | Configuration | | |
| CUSTOMIZATION | Name home | | |
| @ Themes | Description Some details | | |
| | Gateway | | |
| | enp0s31f6 ~ | | |
| | IPv4 | 56107 | |
| | On up | On down | |
| | /usr/sbin/iptables -I FORWARD -I home -j ACCEPT /usr/sbin/iptables -I FORWARD -o home -j ACCEPT /usr/sbin/iptables -t nat -I POSTROUTING -o enp0s31f6 -j MASQUERADE | /usr/sbin/jptables -D FORWARD -1 home -j ACCEPT /usr/sbin/jptables -D FORWARD -0 home -j ACCEPT /usr/sbin/jptables -t nat -D POSTROUTING -0 enp0s31f6-j MASQUERADE | |
| | Public key | Private key 🗞 | |

Fig. 7: Interface's actions and configuration

| | | Â | | | ,h | | | |
|--------------------------------|----------------------------|-------------|---------------|-------------------------------|--------------|-------------------|--------------|------|
| Public key | | | Private key 💐 | | | | | |
| 1gJJbdonldDtW/jlqB2h | nuWPZ8Uhg/UviPuwqoTdUaHo |)= | ••••• | | | | | |
| 🖌 Auto 🚱 | | | | | | | | |
| Save Download | | | | | | | De | lete |
| Information provided at 23:18. | | | | | | | | |
| La Peers + | | | | | | | | |
| Show 10 ¢ entries | we lot entries Search: | | | | | | | |
| Name 11 | Description | IPv4 | Primary DNS | Secondary DNS | NAT 🗈 | Last handshake | Actions | |
| iPhone | None | 10.0.0.3/24 | 8.8.8.8 | None | Yes | None | 2/1 | |
| raspberry pi | None | 10.0.0.2/24 | 8.8.8.8 | None | Yes | None | 2/1 | |
| Showing 1 to 2 of 2 entrie | es | | | | | | Previous 1 N | lext |
| Information provided at 23:18. | | | | | | | | |
| | | | | | | | | |
| L述 Received and transmit | tted data in this session | | | History of received and trans | smitted data | | | |
| 500 B | Received | Transmitted | | 17.6 KB | Rec | eived Transmitted | | |
| 450 B | | | | 15.6 KB | Oct 11, 20 | 21, 10:57:26 pm | 1 | |
| 400 B | | | | 13.7 KB | 15.2 KB | | | |
| 300 B | | | | 11.7 КВ | | • | 1 | |
| 250 B | | | | 9.8 КВ | | | | |

Fig. 8: Interface's peers and traffic data (1)





| Linguard | Ξ | | | | |
|----------|---|---------------|----------------------------------|----------|--|
| SUMMARY | Peer | | | | |
| | Last handshake was 1 minute ago. | | | | |
| | Configuration | | | | |
| | Name | | | | |
| | raspberry pi | | | | |
| | Description | | | | |
| | Some details | | | | |
| | | | | | |
| | Interface | IPv4 | | | |
| | nome | × 10.0.0.2/24 | | | |
| | Primary DNS | Secondary UNS | | | |
| | Public key | Private key 🔊 | | | |
| | w/hwmHMYywoEyxHFDJ0kbFWbaKalR8TXm9j7bpFta0M= | | | | |
| | 🗖 NAT 🥹 | | | | |
| | Save Download | | | | |
| | Information provided at 23:19. | | | | |
| | A Received and transmitted data in this session | | History of received and transmit | ted data | |







| Linguard | = | | | . - |
|------------------------|-------------------------------------|-----------------------------------|--------------------------------------|------------|
| summary 🐵 Dashboard | Settings | | ← B | lack |
| About | 🌣 Web | | | |
| SYSTEM | Maximum login attempts | Secret key Ø 🗞 | Credentials file 🚱 | |
| 😤 Network | 0 | | /var/www/linguard/data/.credentials | |
| ≡ Wireguard | | | | |
| CUSTOMIZATION | Wireguard | | | |
| Settings | Endpoint | Interfaces' directory | Configuration file | |
| Themes | 192.168.0.47 | /var/www/linguard/data/interfaces | /var/www/linguard/data/linguard.yaml | |
| | wg bin | wg-quick bin | iptables bin | |
| | /usr/bin/wg | /usr/bin/wg-quick | /usr/sbin/iptables | |
| | | | | |
| | Logging | | | |
| | Logfile | Log Level | | |
| | /var/www/linguard/data/linguard.log | debug ~ | | |
| | Overwrite 🚱 | | | |
| | | | | |
| | Traffic data collection | | | |
| | ✓ Enabled Ø | | | |

Fig. 12: Settings (1)

| e e un ge | | | |
|----------------|--|-----------------------------------|--------------------------------------|
| Themes | 192.168.0.47 | /var/www/linguard/data/interfaces | /var/www/linguard/data/linguard.yaml |
| - memeo | wg bin | wg-quick bin | iptables bin |
| | /usr/bin/wg | /usr/bin/wg-quick | /usr/sbin/iptables |
| | | | |
| | | | |
| | ST Logging | | |
| | Logfile | Log Level | |
| | /var/www/linguard/data/linguard.log | debug | |
| | Overwrite 🚱 | | |
| | | | |
| | * Traffic data collection | | |
| | | | |
| | Enabled @ | | |
| | Driver | | |
| | JSON | ~ | |
| | Driver configuration | | |
| | | | |
| | "timestamp_tormat": "%d/%m/%Y %H:%M!%S" } | | |
| | | | |
| | | | |
| | | | |
| | Save | | |
| Welcome, admin | | | |
| Linguard 0.2.0 | | | Github - Licen |

Fig. 13: Settings (2)

2.3 In depth

2.3.1 Arguments

The following table describes every argument accepted by Linguard:

| Argu- | Туре | Explanation | Notes |
|----------|----------|---------------------------------------|---|
| ment | | | |
| workdir | Posi- | Path to the Linguard's working direc- | Linguard will store here everything it needs to |
| | tional | tory | work |
| -h help | Optional | Display Linguard's CLI help and exit | |
| debug | Optional | Start the Flask backend in debug | Default value is False |
| | | mode | |

2.3.2 Configuration

Two sample configuration files are provided, uwsgi.sample.yaml and linguard.sample.yaml, although the most interesting one is the second, since the first only contains options for a third party software, UWSGI.

Nonetheless, it is worth noting that the path to the Linguard's working directory (which will be used by Linguard to store stuff) needs to be provided through uwsgi's configuration, using the field pyargv. Moreover, to edit the port and/or the interface in which the web server is running you will need to edit the field http-socket of uwsgi's configuration file.

For now on, we will only discuss Linguard's configuration values. Although the file linguard.sample.yaml contains every possible option, the following tables explain each one of them and detail all possible values.

Logging configuration

These options must be specified inside a logger node.

| Option | Explanation | Values | De- |
|--------|--|---------------------------------------|-------|
| | | | fault |
| level | Set the minimum level of messages to be logged | debug, info, warning, error, fatal | info |
| over- | Whether to overwrite the log file when the application | true, false | false |
| write | starts or not | | |

Web configuration

These options must be specified inside a web node.

| Option | Explanation | Values | Default |
|--------------|--|--------------------|------------------------|
| lo- | Maximum number of login attempts within | (almost) Any inte- | 0 (unlimited attempts) |
| gin_attempt. | s login_ban_time | ger | |
| lo- | Amount of seconds an IP will be banned after too | (almost) Any inte- | 120 |
| gin_ban_tim | emany failed login attempts | ger | |
| secret_key | Key used to secure the authentication process | A 32 characters | A random 32 characters |
| | | long string | long string |

Traffic data collection configuration

These options must be specified inside a traffic node.

| Op- | Explanation | Values | Default |
|-------|-------------------|---|--------------------|
| tion | | | |
| drive | r Driver used to | Any registered driver. You can even craft your own driver us- | The JSON driver, |
| | save and load | ing the base class TrafficStorageDriver. Further information | which stores data |
| | traffic data | will be available through the code documentation | serialized as JSON |
| en- | Whether the | true, false | true |
| ablea | data collection | | |
| | is enabled or not | | |

Note: Linguard will only store the **amount of bytes received and transmitted** by peers, and only if **enabled** is set to true.

Wireguard configuration

These options must be specified inside a wireguard node.

Global options

| Op- | Explanation | Values | Default |
|----------|-----------------------------------|------------------------------------|-------------------------------------|
| tion | | | |
| end- | Endpoint for all peers | Should be something like vpn. | Default value will be your com- |
| point | | example.com, though it may also be | puter's public IP (if it can be ob- |
| | | an IP address | tained) |
| wg_bin | Path to the WireGuard bi- | path/to/file | If not specified, it will be re- |
| | nary file (wg) | | trieved using the whereis com- |
| | | | mand |
| wg_quic | k <u>P</u> atith to the WireGuard | path/to/file | If not specified, it will be re- |
| | quick binary file (| | trieved using the whereis com- |
| | wg-quick) | | mand |
| inter- | Dictionary containing all | A number of interface nodes whose | |
| faces | interfaces of the server | keys are their own UUIDs | |
| ipta- | Path to the iptables binary | path/to/file | If not specified, it will be re- |
| bles_bin | file(iptables) | | trieved using the whereis com- |
| | | | mand |

Interface configuration

These options must be specified inside an interface node.

| Op- | Explanation | Values | Default |
|--------|----------------------------------|----------------------|--|
| tion | | | |
| auto | Whether the interface will | true, false | Default value is true |
| | be automatically brought up | | |
| | when the server starts or not | | |
| de- | A description of the interface | A character string | |
| scrip- | | | |
| tion | | | |
| gw_ifa | centerface used to connect the | A valid network de- | Your computer's default gateway |
| | WireGuard interface to your | vice | |
| | network | | |
| ipv4_a | add Pre4saddress assigned to the | A valid IPv4 address | |
| | interface | | |
| lis- | UDP port used by WireGuard | 1-65535 | |
| ten_po | rto communicate with peers | | |
| name | The interface's name | A character string | It may only contain alphanumeric characters, under- |
| | | | scores and hyphens. It must also begin with a letter |
| | | | and cannot be more than 15 characters long |
| on_up | Linux commands to be exe- | Any linux command | By default, it will add FORWARD and |
| | cuted when the interface is | in path | POSTROUTING rules related to the interface |
| | going to be brought up | - | |
| on_do | whinux commands to be exe- | Any linux command | By default, it will remove FORWARD and |
| | cuted when the interface is | in path | POSTROUTING rules related to the interface |
| | going to be brought down | 1 | |
| peers | Dictionary containing all | A number of peer | |
| | peers of the interface | nodes whose keys | |
| | - | are their own UUIDs | |
| pri- | Private key used to authenti- | A valid private key | |
| vate_k | evate the interface | generated via wg | |
| pub- | Public key used to authenti- | A valid private key | |
| lic_ke | y cate the interface | generated via wg | |
| uuid | Unique identifier | A valid Version 4 | |
| | | UUID | |

Peer configuration

These options must be specified inside an **peer** node.

| Op- | Explanation | Values | Default |
|---------|---|-------------------|--|
| tion | | | |
| dnsl | Main DNS used by the neer | A valid IPv4 ad- | |
| uns1 | Wall Divo used by the peer | dress | |
| drag? | Secondary DNS used by the near | A valid IDv4 ad | |
| unsz | Secondary DNS used by the peer | A vallu if v4 au- | |
| | | uress | |
| ipv4_aa | <i>d</i> Pess 4 address assigned to the peer | A valid IPv4 ad- | |
| | | dress | |
| name | The peer's name | A character | |
| | | string | |
| nat | Linux commands to be executed | Any linux com- | Default value is false. If true, this option |
| | when the interface is going to be | mand in path | will enable the PersistentKeepalive Wire- |
| | brought up | _ | Guard flag |
| pri- | Private key used to authenticate the | A valid private | |
| vate_ke | y peer | key generated | |
| | | via wg | |
| pub- | Public key used to authenticate the | A valid private | |
| lic_key | peer | key generated | |
| | | via wg | |
| uuid | Unique identifier | A valid Version | |
| | _ | 4 UUID | |

2.3.3 Security

Although Linguard stores users' credentials encrypted, it does not implement end-to-end encryption and instead, it relays on TLS to secure the communication between the user and the server. This means you should never run Linguard on its own, but use the https option of uWSGI or set up a reverse proxy if you wish to use plain HTTP with uWSGI. Don't worry, here's how:

uWSGI with HTTPS socket

```
uwsgi:
https: 0.0.0.0:8443,foobar.crt,foobar.key # More info at https://uwsgi-docs.
~readthedocs.io/en/latest/HTTPS.html
master: true
enable-threads: true
chdir: /var/www/linguard
venv: venv
wsgi-file: linguard/__main__.py
pyargv: data
need-plugin: python3
callable: app
die-on-term: true
chmod-socket: 660
vacuum: true
```

Apache reverse proxy

```
<VirtualHost *:443>
ServerName vpn.example.com
ErrorLog ${APACHE*LOG*DIR}/error.log
CustomLog ${APACHE*LOG*DIR}/access.log combined
SSLEngine on
SSLCertificateFile /path/to/crt
SSLCertificateKeyFile /path/to/key
SSLProtocol -all +TLSv1.2 +TLSv1.3
ProxyPreserveHost On
ProxyPass / http://10.0.0.1:8080/
ProxyPassReverse / http://10.0.0.1:8080/
</VirtualHost>
```

Nginx reverse proxy

```
server {
    listen 443;
    server_name vpn.example.com;
    ssl_certificate /path/to/crt;
    ssl*certificate*key /path/to/key;
    ssl_protocols TLSv1.2 TLSv1.3;
    location / {
        proxy*set*header Host $host;
        proxy*set*header X-Real-IP $remote_addr;
        proxy_pass http://10.0.0.1:8080;
    }
}
```

2.4 Contributing

Note: Linguard is and will always be open source.

You may contribute by opening new issues, commenting on existent ones and creating pull requests with new features and bugfixes. Any help is welcome, just make sure you read the following sections, which will guide you to set up the development environment.

2.4.1 Git flow

You should never work directly on the main branch. This branch is only used to gather new features and bugfixes previously merged to the dev branch and publish them in a single package. In other words, its purpose is to release new versions of Linguard.

Hence, the dev branch should always be your starting point and the target of your pull requests.

```
git clone https://github.com/joseantmazonsb/linguard.git
cd linguard
git checkout dev
```

2.4.2 Requirements

You will need to install the following Linux packages:

2.4.3 Dependency management

Poetry is used to handle packaging and dependencies. You will need to install it before getting started to code:

Once you have checked out the repository, you'd install the python requirements this way:

```
poetry config virtualenvs.in-project true
poetry install
```

Then, you would only need to run poetry shell and voilà, ready to code!

Note: Actually, you should always run poetry run pytest before getting started to code in order to check that everything's all right.

2.4.4 Configuration files

Linguard has a setup assistant and does not require you to have an existing configuration file in its working directory. Nonetheless, you may use your own existing file as long as it is valid and named linguard.yaml.

As for the UWSGI configuration, Linguard provides a sample file (uwsgi.sample.yaml) for you to play around with it. Just make sure you run UWSGI using a valid file!

2.4.5 Testing

PyTest and Coverage are used to test Linguard and generate coverage reports, which are uploaded to Codecov.

TDD is enforced. Make sure your code passes the existing tests and provide new ones to prove your new features/bugfixes actually work when making pull requests.

All tests should be anywhere under linguard/tests, and you can run them all using Poetry:

```
poetry run pytest
```

You may as well generate a coverage report using poetry:

poetry run coverage run -m pytest && poetry run coverage report

2.4.6 Building

To build Linguard you may use the build.sh script, which automatically generates a dist folder containing a compressed file with all you need to publish a release.

2.4.7 Versioning

Linguard is adhered to Semantic Versioning.

All releases must follow the format $\{MAJOR\}$. $\{MINOR\}$. $\{PATCH\}$, and git tags linked to releases must follow the format $v\{MAJOR\}$. $\{PATCH\}$. Thus, release 1.0.0 would be linked to the v1.0.0 git tag.

2.4.8 CI/CD

Github Workflows are used to implement a CI/CD pipeline. When pull requests targeting the main or dev branches are opened, a series of tests will automatically be ran to ensure everything is working properly.

Warning: The main branch is used to automatically deploy new releases, and should never be the target of external pull requests.

2.5 Changelog

All notable changes to this project will be documented here.

Note: Linguard is adhered to Semantic Versioning.

2.5.1 1.1.0

What's new

• Ban time is now editable and applies to individual IP addresses instead of globally (which makes much more sense).

Fixes

• Fixed a bug with the settings page which caused the display of default/last saved settings everytime the page was reloaded, even though the values were actually being stored in the configuration file and applied.

Docs

• Added entry for ban time.

2.5.2 1.0.1

Fixes

• Fixed a bug related to versioning which caused the app to start in dev mode.

Docs

• Removed "Versions" empty section from index.

2.5.3 1.0.0

What's new

- QR codes! You can scan a QR code to get the WireGuard configuration of any peer or interface.
- Docker is finally here! For now on, there will be official docker images available for every release.
- Display the IP address of the interface to be used when adding or editing a peer.
- Updating the name of an interface also updates all references inside the "On up" and "On down" text areas.
- Delete buttons have been relocated in the Interface and Peer views.

Fixes

- Fixed a bug when updating the username or password which made the "Logged in {time} ago" sign show no time at all.
- Removed the possibility to add peers if there are no WireGuard interfaces.
- Ensured that peers can only be assigned valid, unused and not reserved IP addressed.
- Ensured that peers' IP addresses are in the same network of their interface.
- Ensured that interfaces can only be assigned valid, unused and not reserved IP addressed.

- Ensured that interfaces' cannot be assigned an IP address belonging to a network which already has an interface.
- Fixed a bug when updating an interface's gateway, which only updated one appearance of the previous gateway in the "On up" and "On down" text areas.
- Fixed the behaviour of the overwrite flag regarding the logging settings which was causing to overwrite the log file each time the settings were saved instead of every time Linguard boots up.

Docs

- Improved documentation about the development environment.
- Fixed a bunch of typos.
- Fixed the Traffic Data Driver table.

2.5.4 0.2.0

- Easy first time setup, which automatically detects the location of the required binaries and sets the public IP as endpoint by default.
- Everything in one place: workdir-based architecture.
- Removed option to log to standard output.
- Includes a ready-to-go uWSGI configuration file.
- Removed the linguard.sample.yaml file in favour of the first time setup.
- Settings are now accessible through the side navbar.

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search