Wirow

1. Setting Up

1.1. General Prerequisites

`Linux x86_64` server instance accessible by valid domain name (DNS) from the Internet.

1.2. Minimal Hardware Requirements for server

- 2 CPU/vCPU cores
- 4 GB RAM
- SSD storage

> 🔥 The CPU can be high if you are using the video recording function in rooms.

1.3. Requirements for Wirow client software

Supported browsers

- Chrome 74+
- Firefox 70+
- Safari 14+

1.4. Network accessibility Checklist

- A firewall allows all outgoing network connections as the server must have access to the WebRTC clients and the license server.
- The firewall allows inbound TCP/UDP connections to the following ports
  - HTTP 80, HTTPS 443
  - WebRTC RTP ports 10000..59999

A `wirow` executable must be run by a non-root user and allowed to listen on 80 and 443 network ports. To do this, simply run the following `setcap` command:

```
sudo /usr/sbin/setcap 'cap_net_bind_service=+ep' ./wirow
```

1.5. Domain Name (DNS)

Wirow server must be accessible by `https` protocol. This is a mandatory requirement. Thus, you need to point your domain registrar's DNS server to the actual IP address of the Wirow server.

1.6. Running the Wirow Server

```
Usage: ./wirow [options]

  -c <cfg>            .ini configuration file
  -d <dir>            Data files directory
  -n <domain>         Domain name used to obtain Let's Encrypt certs
  -l <ip>[@<pub ip>]  Listen IP or IP mapping if server behind NAT
  -p <port>           Server network port number
  -a <password>       Set the specified password for `admin` account
  -s                  The server runs behind an HTTPS proxy
  -t                  Clear database data on start
  -v                  Show version and license information
  -h                  Show this help message
```

> ℹ️ Use `-a` option to set an initial password for `admin` account at first run. Later in `Admin UI` you may add other users and change your password.

1.7. Wirow server with Real IP Address

```
./wirow -n <domain name>
```

Example:

```
./wirow -n conferences.mycompany.com
```

In this case, Wirow automatically installs Let's Encrypt HTTPS certificates for `https://conferences.mycompany.com`

1.8. Wirow server behind NAT

```
./wirow -n <domain name> -l '<private ip>@<public ip>'
```

1.9. Wirow server behind an HTTP proxy

> ⚠️ We do not recommend running Wirow behind an HTTP proxy, as this will break one of the strongest features of the product — the ease of installation and server configuration.

> ⚠️ Please keep in mind — Wirow WebRTC RTP ports (usually in range `10000..59999`) must be accessible from external network even behind an HTTP proxy. So it is wrong to bind the server to `localhost` behind the proxy.

Example of Apache2 Proxy Configuration

```
<VirtualHost *:443>
    SSLCertificateFile /etc/letsencrypt/live/<domain name>/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/<domain name>/privkey.pem
    Include /etc/letsencrypt/options-ssl-apache.conf

    ProxyRequests          Off
    ProxyPreserveHost      On

    ProxyPass       /ws/channel  ws://<wirow ip>:8080/ws/channel
    ProxyPassReverse /ws/channel  ws://<wirow ip>:8080/ws/channel

    ProxyPass       /            http://<wirow ip>:8080/
    ProxyPassReverse /            http://<wirow ip>:8080/

    <Location "/">
        RequestHeader set X-Forwarded-Proto "https"
        RequestHeader set X-Forwarded-Port "443"
    </Location>
</VirtualHost>
```

```
a2enmod ssl proxy proxy_http proxy_wstunnel
```

```
./wirow -s -p 8080
```

Example of NGINX Proxy Configuration

```
server {
  server_name     <domain name>;
  listen 443 ssl;
  ssl_certificate /etc/letsencrypt/live/<domain name>/fullchain.pem;
  ssl_certificate_key /etc/letsencrypt/live/<domain name>/privkey.pem;
  include /etc/letsencrypt/options-ssl-nginx.conf;
  ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

  location /ws/channel {
    proxy_pass http://<wirow ip>:8080/ws/channel;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
  }
  location / {
    proxy_pass      http://<wirow ip>:8080/;
    proxy_redirect  default;
  }
}
server {
  server_name     <domain name>;
  listen  80;
  if ($host = <domain name>) {
    return 301 https://$host$request_uri;
  }
  return 404;
}
```

```
./wirow -s -p 8080
```

2. Wirow .ini Configuration

Additional Wirow server parameters can be specified in the `.ini` configuration file, as shown in the example below.

```
./wirow ... -c ./wirow.ini ...
```

2.1. Example of wirow.ini Config

The configuration file can be specified by `-c` option

```
./wirow -c <config.ini>
```

```
;; Wirow example configuration.
```

```
;;
;; Any part of configuration may contain placeholders replaced by
;; runtime values:
;;
;;  {home}            Path to user home directory.
;;  {cwd}             Current working directory of wirow process.
;;  {config_file_dir}   Path to directory where configuration file resides.
;;  {programm}         Path to wirow executable.
;;

[main]

;; IP address to listen.
;;  auto - server will autodetect IP address to listen.
;; Overridden by `-l <ip>[@<pub ip>]` command line option
host = auto

;; HTTP/HTTPS listen port.
;; If cert_file / cert_key_file / domain_name specified this
;; port will be used for HTTPS traffic.
;; Overridden by `-p <port>` command line option
;;
;; Example:
port = 8888

;; DNS domain name used for server in order to obtain Let's Encrypt TLS
certificate.
;; Overridden by `-n <domain>` command line option
;;
;; Example:
domain_name = foo.example.com

;; HTTP port used to redirect user to HTTPS protocol.
;; Also HTTP used to pass ACME challenge during process of generating Let's
Encrypt TLS certificates.
https_redirect_port = 80

;; Data directory where database files resides
data = {cwd}

;; Path to x509 PEM certificate and key file for TLS layer
;;
;; Example:
cert_file = {config_file_dir}/cert.pem
cert_key_file = {config_file_dir}/key.pem

;; Stun / turn servers
[servers]

;; Stun and turn servers
;;
```

```
;; Example:
turn_servers = user:password@host
stun_servers = stun.l.google.com:19305 stun1.l.google.com:19305
stun2.l.google.com:19305

;; RTC / WebRTC options
[rtc]

;; WebRTC RTP ports range
ports = 10000..59999

;; Mapping <private ip> to <public ip> used for server behind NAT
;;
;; `auto` - Means webrtc server endpoint will listen on autodetected
;;
;; Example:
;;   listen_announced_ips = 0.0.0.0@192.168.1.37
listen_announced_ips = auto
```