Admin Panel aktivieren

> It's heavily recommended to activate HTTPS before enabling this feature, to avoid possible MITM attacks.

This page allows a server administrator to view all the registered users and to delete them. It also allows inviting new users, even when registration is disabled.

To enable the admin page, you need to set an authentication token. This token can be anything, but it's recommended to use a long, randomly generated string of characters, for example running `openssl rand -base64 48`.

Keep this token secret, this is now the password to access the admin area of your server! Which is why you should secure the admin token.

To set the token, use the `ADMIN_TOKEN` variable:

```
docker run -d --name vaultwarden \
  -e ADMIN_TOKEN=some_random_token_as_per_above_explanation \
  -v /vw-data/:/data/ \
  -p 80:80 \
  vaultwarden/server:latest
```

After this, the page will be available in the `/admin` subdirectory.

The first time you save a setting in the admin page, `config.json` will be generated in your `DATA_FOLDER`. Values in this file will take precedence over the corresponding environment variable.

Note that config changes in the admin page do not take effect until you click the `Save` button. For example, if you are testing SMTP settings, and you change the `SMTP Auth mechanism` setting and then click `Send test email` to test the change, this won't work as expected -- since you didn't click `Save`, the `SMTP Auth mechanism` change won't have taken effect.

Note: After changing the `ADMIN_TOKEN`, any admins that are currently logged in will still be able to use their existing login sessions until expiration. The admin session lifetime is configurable, with a default of 20 minutes.

Disabling the admin page

In order to disable the admin page you have to unset the `ADMIN_TOKEN` and restart Vaultwarden.

Note: Removing the environment variable `ADMIN_TOKEN` won't disable the admin page if the value is persisted in the `config.json` file mentioned above. **To disable admin page**, make sure no `ADMIN_TOKEN` environment variable is set, and no `"admin_token"` key exists in `config.json`, if that file exists.

Secure the `ADMIN_TOKEN`

> This feature is available since 1.28.0+.

> Using environment variables is preferred.
> But if you updated settings via the admin interface you need to update the admin token via the same web interface!
> Please **do not** edit the `config.json` manually since that could cause issues if done wrong

> To log into the admin page after securing the token, you instead use the password provided during token creation.

Previously the `ADMIN_TOKEN` could only be in a plain text format.

You can now hash the `ADMIN_TOKEN` using Argon2 by generating a PHC string.
This can be generated by using a built-in `hash` command within Vaultwarden, or use the `argon2` CLI tool.

Within the vaultwarden application we have two presets, one using the Bitwarden defaults, and one using the OWASP recommendations.

> If you keep getting the message `You are using a plain text ADMIN_TOKEN which is insecure.`, then you either saved the settings via the admin interface already, and environment variables will not be used. Or you need to verify if you used the correct format. Carefully read the **How to prevent variable interpolation in `docker-compose.yml`** section below.

Some examples on how to generate an Argon2id PHC hash.

Using `vaultwarden hash`

There is a PHC generator built-in into Vaultwarden which you can run via the CLI `vaultwarden hash`.
This can be done via `docker exec` on the already running instance, or by running this locally via

docker on your own system.
I use `vwcontainer` as the container name below, replace this with the correct container name of your instance.
The Vaultwarden CLI will ask for the password twice, and if both are the same it will output the generated PHC string.

Examples:

```
# Using the Bitwarden defaults (default preset)
# Via docker on a running container
docker exec -it vwcontainer /vaultwarden hash

# Via docker and creating a temporary container
docker run --rm -it vaultwarden/server /vaultwarden hash

# Using the vaultwarden binary directly
./vaultwarden hash

# Using the OWASP minimum recommended settings
# Via docker on a running container
docker exec -it vwcontainer /vaultwarden hash --preset owasp

# Via docker and creating a temporary container
docker run --rm -it vaultwarden/server /vaultwarden hash --preset owasp

# Using the vaultwarden binary directly
./vaultwarden hash --preset owasp
```

Using `argon2`

You can also use the `argon2` CLI available on most Linux Distro's.

```
# Using the Bitwarden defaults
echo -n "MySecretPassword" | argon2 "$(openssl rand -base64 32)" -e -id -k 65540 -t 3 -p 4
# Output:
$argon2id$v=19$m=65540,t=3,p=4$bXBGMENBZUVzT3VUSFErTzQzK25Jck1BN2Z0amFuWjdSdVlIQVZqYzAzYz0$T9m
73OdD2mz9+aJKLuOAdbvoARdaKxtOZ+jZcSL9/N0

# Using the OWASP minimum recommended settings
echo -n "MySecretPassword" | argon2 "$(openssl rand -base64 32)" -e -id -k 19456 -t 2 -p 1
```

```
# Output:
$argon2id$v=19$m=19456,t=2,p=1$cXpKdUxHSWhlaUs1QVVsSStkbTRPQVFPSmdpamFCMHdvYjVkWTVKaDdpYz0$E1U
gBKjUCD2Roy0jdHAJvXihugpG+N9WcAaR8P6Qn/8
```

Use this string in your docker/podman CLI command. For `docker-compose.yml` files follow the instructions below. If you are on an existing setup, do not forget to update your password/token via the admin web interface, too.

How to prevent variable interpolation in `docker-compose.yml`

When [[using Docker Compose]] and you configure the `ADMIN_TOKEN` via the `environment` directive you need to escape all five occurrences of the dollar sign `$` in the generated argon2 PHC string using two dollar signs `$$` in order to prevent variable interpolation:

```
  environment:
    ADMIN_TOKEN:
$$argon2id$$v=19$$m=19456,t=2,p=1$$UUZxK1FZMkZoRHFQRlVrTXZvS0E3bHpNQW55c2dBN2NORzdsa0Nxd1JhND0
$$cUoId+JBUsJutlG4rfDZayExfjq4TCt48aBc9qsc3UI
```

This can be done automatically e.g. using sed by adding `| sed 's#\$#\$\$#g'` to the end of the `argon2` command line above.

Otherwise you'll get warning messages and the variable will not be set correctly:

```
WARNING: The argon2id variable is not set. Defaulting to a blank string.
WARNING: The v variable is not set. Defaulting to a blank string.
WARNING: The m variable is not set. Defaulting to a blank string.
...
```

> This is not the case when using a `.env` file for `docker-compose.yaml`
> As shown below. In this case just use the single `$` variant.
> The same for using the docker/podman cli using `-e ADMIN_TOKEN`.

```
/docker-data
├── .env
├── docker-compose.yaml
├── vaultwarden/data
```

.env:

Make sure you use single quotes in the `.env` file used by docker-compose.

```
VAULTWARDEN_ADMIN_TOKEN='$argon2id$v=19$m=65540,t=3,p=4$MmeK.....'
```

docker-compose.yaml:

```
services:
  vaultwarden:
    image: ghcr.io/dani-garcia/vaultwarden
    container_name: vaultwarden
    restart: unless-stopped
    volumes:
      - /path/to/vaultwarden/data/:/data/
    environment:
      - ADMIN_TOKEN=${VAULTWARDEN_ADMIN_TOKEN}
```

You can check your configuration by calling `docker compose config`, you should see the escaped $-sign as double-$.

Revision #3
Created 2024-07-28 17:50:15 UTC by CoreStream Group
Updated 2024-07-28 17:57:52 UTC by CoreStream Group