Author	Thiébaud Modoux (Pryv)
Reviewer	Loïc Correvon (Pryv)
Date	29.11.2019
Version	1

Pryv.io MFA

Multi-factor authentication micro-service

Summary

This document describes a Pryv.io micro-service that allows to activate multi-factor authentication (MFA) on top of Pryv.io login calls. Once MFA is activated for a given Pryv.io user, subsequent login calls will enforce the completion of the MFA flow (challenge, verification) before obtaining the Pryv.io personal access token.

Table of contents

Installation

The MFA micro-service is meant to be installed on each core machine, by adding the service-mfa Docker image to the core Dockerfile, as follows:

- **Single node setup**:

```
# pryv.yml

  mfa:
    image: "pryvsa-docker-release.bintray.io/pryv/service-mfa:0.0.7"
    container_name: service_mfa
    networks:
      - frontend
    volumes:
      - ./pryv/mfa/conf/:/app/conf/:ro
      - ./pryv/mfa/log/:/app/log/
    links:
      - core
    restart: always
```

- **Cluster setup**:

```
# core.yml

  mfa:
    image: "pryvsa-docker-release.bintray.io/pryv/service-mfa:0.0.7"
    container_name: service_mfa
    networks:
      - frontend
    volumes:
      - ./core/mfa/conf/:/app/conf/:ro
      - ./core/mfa/log/:/app/log/
    links:
      - core
    restart: always
```

Then, the Nginx container on each core machine needs to be configured to forward login and MFA calls to the MFA micro-service. Adapt the Nginx configuration on core(s) with the following:

```
# nginx/conf/site-443.conf

upstream mfa_server {
  server mfa:7000 max_fails=3 fail_timeout=30s;
}

...

# API (Core)
server {

  ...

  # MFA
  location /auth/login {
    proxy_pass http://mfa_server/login;
  }

  location /mfa/ {
    proxy_pass http://mfa_server/mfa/;
```

```
    }

    ...

  }
```

Configuration

The MFA micro-service can be configured in the file **mfa/conf/service-mfa.conf** (the exact path depends on the Docker volumes defined above).

Here is the default configuration as example:

```
    http: {
      port: 7000,
      ip: '127.0.0.1',
    },
    logs: {
      prefix: 'service-mfa',
      console: {
        active: true,
        level: 'info',
        colorize: true
      },
      file: {
        active: false
      },
    },
    // Pryv.io core to which the login calls will be forwarded
    core: {
      url: 'http://core_router:1337'
    },
    // API to send MFA challenge by SMS
    sms: {
      endpoints: {
        challenge: '', // Endpoint that triggers the MFA challenge
        verify: '', // Endpoint that verifies the MFA challenge
      },
      auth: '' // API key, sent as 'Authorization' header
    },
    // Sessions are used to cache the state of MFA processes in progress
    sessions: {
      ttlSeconds: 1800 // Duration in seconds after which sessions are destroyed
    }
```

The endpoints defined as **sms:endpoints:challenge** and **sms:endpoints:verify** should correspond to the SMS authentication API of your choice, which will generate and send, respectively verify, the MFA challenge.

If required, the key defined as **sms:auth** will be provided to the SMS authentication API as **Authorization** header.

MFA flow

Activation

1) POST /mfa/activate
Begin the MFA activation flow for a given Pryv.io user. It will trigger the MFA challenge (via the SMS authentication API).

Request body:
Since this call will be forwarded to the SMS authentication API, the request body should match the one expected by the endpoint configured above as **sms:endpoint:challenge** (e.g. **phone_number**: the phone number to which the SMS code will be sent).

Request headers:

- **Authorization**: Pryv.io personal token (retrieved from a preliminary login call).

Response:

```
HttpCode: 302
Body: { "mfaToken": "mfaToken" }
```

Example:

```
curl -i -X POST -H 'Authorization: pryvPersonalToken' -H 'Content-Type: application/json' \
-d '{"phone_number":"41791234567"}' https://testuser.pryv.me/mfa/activate
```

2) POST /mfa/confirm
Confirm the MFA activation by verifying the MFA challenge (via the SMS authentication API).

Request body:
Since this call will be forwarded to the SMS authentication API, the request body should match the one expected by the endpoint configured above as **sms:endpoint:verify** (e.g. **code**: the SMS code to be verified).

Request headers:

- **Authorization**: mfaToken

Response:

```
HttpCode: 200
Text: "MFA activated."
```

Example:

```
curl -i -X POST -H 'Authorization: mfaToken' -H 'Content-Type: application/json' \
-d '{"code":"1234"}' https://testuser.pryv.me/mfa/confirm
```

Login

1) POST /auth/login
Proxy the Pryv.io login call, performing MFA authentication if needed.

Since this call will be forwarded to the Pryv.io API, the request body and headers should match the ones expected by the login method. It will begin the MFA verification flow if activated, or just return the Pryv.io personal access token otherwise.

Request body:
- **username**: Pryv.io username
- **password**: Pryv.io password
- **appId**: app's unique identifier

Request headers:
- **Origin** (or **Referer**): see Trusted apps verification

Response:
- if MFA activated:

```
HttpCode: 302
Body: { "mfaToken": "mfaToken" }
```

- if MFA not activated:

```
HttpCode: 200
Body: { "token": "pryvPersonalToken" }
```

Example:
```
curl -i -X POST -H 'Origin: https://sw.pryv.me' -H 'Content-Type: application/json' \
-d '{"username":"testuser","password":"testpassword","appId":"my-app-id"}' \
https://testuser.pryv.me/auth/login
```

2) POST /mfa/challenge
Trigger the MFA challenge (via the SMS authentication API).

Request headers:
- **Authorization**: mfaToken

Response:
```
HttpCode: 200
Text: "Please verify MFA challenge."
```

Example:
```
curl -i -X POST -H 'Authorization: mfaToken' https://testuser.pryv.me/mfa/challenge
```

3) POST /mfa/verify

Verify the MFA challenge (via the SMS authentication API).

Request body:

Since this call will be forwarded to the SMS authentication API, the request body should match the one expected by the endpoint configured above as **sms:endpoint:verify** (e.g. **code**: the SMS code to be verified). It will return the Pryv.io personal access token if the verification succeeded.

Request headers:

- **Authorization**: mfaToken

Response:

```
HttpCode: 200
Body: { "token": "pryvPersonalToken" }
```

Example:

```
curl -i -X POST -H 'Authorization: mfaToken' -H 'Content-Type: application/json' \
-d '{"code":"1234"}' https://testuser.pryv.me/mfa/verify
```