Управление вторым фактором входа

Оглавление

Основная информация

В стандартных сценариях RooX UIDM есть поддержка второго фактора аутентификации, при этом есть гибкие механизмы управления вторым фактором, позволяющие, как администратору решения, так и конечному пользователю определять, когда второй фактор должен использоваться, а когда нет.

Так, в RooX UIDM выделены следующие возможности управления вторым фактором:

- персональные настройки пользователя. Дают возможность пользователю управлять необходимостью второго фактора в разных условиях.
- настройки для приложения. Позволяют включать или выключать необходимость второго фактора для отдельного интегрированного с RooX UIDM приложения. Например, можно включить второй фактор для Web-приложения и выключить для сервиса, реализующего machine-2-machine сценарии.
- глобальная настройка по умолчанию. Задает поведение решения, если не задана настройка для конкретного пользователя или приложения.

Настройки управления вторым фактором входа

Персональные настройки пользователя

Персональные настройки пользователя сохраняются в БД RooX UIDM в таблице OtpPrincipalConfiguration. Для каждого пользователя в этой таблице могут содержаться следующие ключи:

- `otp.social.mapping.login.enabled` — включен ли OTP для входа через социальную сеть;
- `otp.social.mapping.attach.enabled` — включен ли OTP для создания связки с аккаунтом социальной сети.
- `otp.social.mapping.reattach.enabled` — включен ли OTP для замены связки с аккаунтом социальной сети.

- `otp.login.enabled` — включен ли OTP для входа по логину и паролю;
- `otp.action.enabled` — включен ли OTP под операцию.

Для управления этими настройками в состав решения RooX UIDM входит специальный микросервис, предоставляющий защищенное API для управления настройками как от имени пользователя, так и от имени системы. Подробнее про установку и настройку сервиса написано далее.

Настройки приложения

Настройки для отдельных приложений задаются в env.properties в следующих параметрах:

```
com.rooxteam.uidm.<category_name>.otp.enabled_for_client_id=<client_id1>=<value for client_id1>, <client_id2>=<value for client_id2>, ...
```

где `<category_name>` - одна из следующих категорий:

- otp-login - вход с поддержкой второго фактора
- login-by-otp - вход по одноразовому паролю
- social-auth - вход через социальную сеть
- social-attach - привязка аккаунта социальной сети
- social-reattach - перепривязка аккаунта социальной сети к другому пользователю
- otp-sign - подпись документа
- change-msisdn - смена номер телефона
- msisdn-password-recovery - восстановление пароля по номеру телефона
- otp-verify - подтверждение email или номера телефона

a <client_idN> - имена соответствующих client_id <value for client_idN> - может принимать значения `true` или `false`

Глобальная настройка по умолчанию

Глобальная настройка задается в env.properties в следующей настройке (значение приведено для примера). В настройке через запятую перечисляются пары `ключ=значение` для всех флагов

```
# описание: Значения по умолчанию для OTP настроек
# тип данных: dictionary
com.rooxteam.uidm.otp.code.enabled.default=otp.login.enabled=false,otp.social.mapping.login.enabled=false
```

Настройка решения для управления пользовательскими настройками

Для управления пользовательскими настройками существует отдельный микросервис. Для его использования необходимо:

1. установить микросервис
2. настроить параметры микросервиса

3. настроить правила маршрутизации запросов на микросервис
4. настроить политики RooX UIDM для авторизации запросов на микросервис

Установка микросервиса

Установите пакет `webapi-server-otp-settings-10`. После установки в системе появится новый сервис `roox-webapi-otp-settings-10`

ЗАМЕТКА

сервис может быть предоставлен в виде docker container. Обратитесь в поддержку за дополнительной информацией.

Настройка параметров микросервиса

Для корректной работы микросервиса в файле настроек env.properties должны присутствовать следующие параметры:

Настройки подключения к БД

Ниже приведен пример настроек для подключения к Oracle DB.

ЗАМЕТКА

если компонент устанавливается на тот же сервер, где уже установлен сервис roox-sso, дополнительно ничего добавлять в настройки не требуется.

```
# описание: JDBC драйвер стандартной СУБД WebAPI
# тип данных: string
# единицы измерения: нет
# ограничения: полное имя класса JDBC драйвера
com.rooxteam.webapi.database.driver=oracle.jdbc.driver.OracleDriver

# описание: uri к стандартной СУБД WebAPI
# тип данных: string
# единицы измерения: нет
# ограничения: существующий uri, формат jdbc url
com.rooxteam.webapi.database.url=jdbc:oracle:thin:@//uidm-db.rooxteam.com:1521/XE

# описание: имя пользователя стандартной СУБД WebAPI
# тип данных: string
# единицы измерения: нет
# ограничения: существующий пользователь в СУБД
com.rooxteam.webapi.database.user=RX_OAUTH

# описание: пароль пользователя стандартной СУБД WebAPI
# тип данных: string
# единицы измерения: нет
# ограничения: существующий пользователь в СУБД
com.rooxteam.webapi.database.password=RX_OAUTH

# описание: максимальное количество connection в пуле
# тип данных: integer
# единицы измерения: нет
# ограничения: 1 - 2^31-1
com.rooxteam.webapi.database.connection.max=5
```

```
# описание: запрос для проверки доступности СУБД
# тип данных: string
# единицы измерения: нет
# ограничения: валидный запрос для соответствующей СУБД
com.rooxteam.webapi.database.check.query=SELECT 1 FROM DUAL

# описание: включить монитор пула БД
# тип данных: boolean
# единицы измерения: нет
# значение по умолчанию: false
com.rooxteam.webapi.database.monitor.enabled=false

# описание: Время ожидания вызова getConnection
# тип данных: number
# единицы измерения: миллисекунды
# ограничения: нет
# по умолчанию: 0
com.rooxteam.webapi.database.connection_timeout=10000
```

Настройки подключения к RooX UIDM

Для обработки запросов от пользователей сервису необходимы настройки AAL для проверки токена пользователя. Ниже приведен пример настроек.

```
# описание: Базовый URL RooX UIDM sso-server для валидации токенов пользователя
# тип данных: string
# единицы измерения: нет
# ограничения: валидный URL
com.rooxteam.aal.sso.endpoint=https://uidm.demo.rooxteam.com/sso

# описание: Имя агента (client id) OAuth для сервиса
# тип данных: string
# единицы измерения: нет
# ограничения: существующий Client ID
com.rooxteam.aal.auth.client=otp-settings-service

# описание: Секретный ключ (client secret) агента OAuth ID для сервиса
# тип данных: string
# единицы измерения: нет
# ограничения: нет
com.rooxteam.aal.auth.password=password
```

Настройка правил маршрутизации

Для работы микросервиса необходимо, чтобы запросы пользователей, отправленные на соответствующий URL `/otp-settings-1.0` попадали на сервис. Сервис принимает запросы по http на порту 29108.

Если непосредственно на сервере используется haproxy для маршрутизации запросов, необходимо добавить в него следующие настройки:

Настройки секции frontend

```
acl url_otp_sett_webapi     path_beg /otp-settings-1.0/
acl url_otp_sett_webapi     path /otp-settings-1.0
```

```
use_backend otp-settings    if url_otp_sett_webapi
```

Настройки backend (добавление нового)

```
backend otp-settings
    balance     roundrobin
    server      otp-settings 127.0.0.1:29108 check
```

Настройка политик доступа

При получении запроса на управление пользовательскими параметрами, `webapi-otp-settings` обращается на RooX UIDM для проверки прав пользователя. Для этого надо в файл конфигурации `openam_tune_batch_2.conf` добавить строку: `create-policies --realm / -X /opt/roox-sso-01/conf/OtpSettingsPolicy.xml` и создать файл `/opt/roox-sso-01/conf/OtpSettingsPolicy.xml` со следующим содержимым:

```
<?xml version="1.0"?>
<!DOCTYPE Policies
        PUBLIC "-//OpenSSO Policy Administration DTD//EN"
        "jar://com/sun/identity/policy/policyAdmin.dtd">
<Policies>
    <Policy name="otp-settings-single" referralPolicy="false" active="true">
        <Rule name="urlScopeRule">
            <ServiceName name="iPlanetAMWebAgentService"/>
            <ResourceName name="/otp-settings/:id/otp/settings/:param"/>
            <AttributeValuePair>
                <Attribute name="GET"/>
                <Value>allow</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="POST"/>
                <Value>allow</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="PUT"/>
                <Value>allow</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="DELETE"/>
                <Value>allow</Value>
            </AttributeValuePair>
        </Rule>
        <Subjects>
            <Subject name="AuthUsers" type="AuthenticatedUsers"/>
        </Subjects>
        <Conditions>
            <Condition name="spelCondition" type="SpelTokenCondition">
                <AttributeValuePair>
                    <Attribute name="allow-if"/>
                    <Value>token['claims'].containsKey('roles') and token['claims']
['roles'].contains('ROLE_SYSTEM') or token['sub'] == #principalId or token['sub'] ==
'@me'</Value>
                </AttributeValuePair>
                <AttributeValuePair>
                    <Attribute name="denied-advice"/>
                    <Value>Forbidden</Value>
                </AttributeValuePair>
```

```
            </Condition>
        </Conditions>
    </Policy>
    <Policy name="otp-settings-batch" referralPolicy="false" active="true">
        <Rule name="urlScopeRule">
            <ServiceName name="iPlanetAMWebAgentService"/>
            <ResourceName name="/otp-settings/:id/otp/settings"/>
            <AttributeValuePair>
                <Attribute name="GET"/>
                <Value>allow</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="PATCH"/>
                <Value>allow</Value>
            </AttributeValuePair>
        </Rule>
        <Subjects>
            <Subject name="AuthUsers" type="AuthenticatedUsers"/>
        </Subjects>
        <Conditions>
            <Condition name="spelCondition" type="SpelTokenCondition">
                <AttributeValuePair>
                    <Attribute name="allow-if"/>
                    <Value>token['claims'].containsKey('roles') and token['claims']
['roles'].contains('ROLE_SYSTEM') or token['sub'] == #principalId or token['sub'] ==
'@me'</Value>
                </AttributeValuePair>
                <AttributeValuePair>
                    <Attribute name="denied-advice"/>
                    <Value>Forbidden</Value>
                </AttributeValuePair>
            </Condition>
        </Conditions>
    </Policy>
</Policies>
```