

Как настроить авторизацию пользователей в веб- интерфейсе установщика

Инструкция для администраторов

Оглавление

| | |
|--|---|
| Типы авторизации в установщике | 3 |
| Настройки авторизации | 3 |
| Настройка OIDC сервера | 3 |
| Настройка авторизации по IP | 5 |
| Настройка прокси-авторизации | 6 |
| Переопределение атрибутов пользователей | 6 |
| Дополнительные настройки авторизации | 7 |
| Пример полной конфигурации | 8 |
| Как получить логи авторизации через OIDC-провайдер | 8 |
| Как закрыть доступ до установщика с помощью TLS | 9 |

Типы авторизации в установщике

1. Сопоставление IP-пользователь — пользователь определяется по IP адресу.
2. Прокси-авторизация. Предполагается, что установщик работает за прокси-сервером, который занимается авторизацией, а установщик получает данные о пользователе из заголовков.
3. Через OIDC сервер. Пользователь перенаправляется к OIDC серверу для авторизации.

Типы авторизации перечислены в порядке убывания приоритета. Все 3 типа авторизации могут быть настроены одновременно. Например, если данные о пользователе будут получены из заголовков, то перенаправления к OIDC серверу не будет. Чтобы отключить какой-то из типов авторизации, достаточно не указывать его в файле конфигурации.

Настройки авторизации

Вся настройка производится в файле конфигурации `deployerParams.yaml`. По умолчанию файл находится в папке `/home/deployer/`. Файл содержит 4 основных блока:

- `oidc` — блок настройки OIDC сервера.
- `byIpUsers` — блок настройки авторизации по IP. Если пользователь подключится с заданного IP адреса, то ему будет назначено заданное имя пользователя. Сделано для подключения сторонних систем, например, ботов или CI/CD.
- `proxyAuth` — блок настройки прокси-авторизации: когда авторизацией занимается вышестоящий сервис, а в установщик приходят только заголовки с уже определёнными значениями.
- `overrideUsers` — переопределение атрибутов пользователей. Эти значения имеют приоритет перед данными, полученными из OIDC или заголовков.

В установщике есть 2 роли пользователей:

- **admin** — полный доступ.
- **viewer** — только чтение.

Внимание

Никакие из приведённых выше настроек не являются значениями по умолчанию. По умолчанию все поля отсутствуют в файле конфигурации, а установщик работает без авторизации.

Настройка OIDC сервера

Содержит следующие поля:

| Название поля | Описание | Пример значения |
|--------------------------|--|---|
| host | URL OIDC сервера | <code>https://biz.wm1.on-premise.ru/auth/realms/EXCH.ON-PREMISE.RU/protocol/openid-connect</code> |
| skipTls | Игнорировать ошибку сертификата при подключении к OIDC серверу | <code>false</code> |
| clientId | ID OIDC клиента | <code>api</code> |
| clientSecret | Секрет OIDC клиента | <code>otBrIQrxqvXj9VUt0kR1TcuBsmDzx8</code> |
| authPath | Путь API клиентской авторизации | <code>/auth</code> |
| tokenPath | Путь API работы с токенами | <code>/token</code> |
| infoPath | Путь API информации о пользователе | <code>/userinfo</code> |
| scope | OAuth-scope OIDC сервера | <code>email</code> |
| disableRefresh | Не проверять актуальность OAuth токена по истечении срока действия | <code>false</code> |
| infoFields | Мапинг полей в ответе OIDC API информации о пользователе | |
| infoFields: email | Содержит email пользователя в ответе OIDC API информации о пользователе | <code>email</code> |
| infoFields: username | Обязательное поле — содержит имя пользователя в ответе OIDC API информации о пользователе | <code>preferred_username</code> |
| infoFields: firstName | | <code>given_name</code> |

| Название поля | Описание | Пример значения |
|----------------------|---|-----------------|
| | Содержит имя в ответе OIDC API информации о пользователе | |
| infoFields: lastName | Содержит фамилию в ответе OIDC API информации о пользователе | family_name |
| infoFields: role | Содержит роль пользователя в ответе OIDC API информации о пользователе. Если окажется пустым, то будет подставлена defaultRole (описана ниже) | deployerAccess |

Пример конфигурации:

```
auth:
  oidc:
    host: https://biz.ws.vkwm1.on-premise.ru/auth/realms/EXCH.ON-PREMISE.RU/protocol/openid-connect
    skipTls: false
    clientId: api
    clientSecret: otBrIQrxqvXj9VUt0k5iR1TcuBsmDzx8
    authPath: /auth
    tokenPath: /token
    infoPath: /userinfo
    scope: email
    disableRefresh: false
    infoFields:
      email: email
      username: preferred_username
      firstName: given_name
      lastName: family_name
      role: deployerAccess
```

Настройка авторизации по IP

Содержит следующие поля:

- `cidr` — IP/Subnet клиента.
- `username` — **Обязательное поле**. Идентификатор (имя пользователя), который будет назначен клиенту. Атрибуты пользователя будут взяты из блока `overrideUsers`.

Пример конфигурации:

```
auth:
  byIpUsers:
```

```
- cidr: 172.20.70.190
  username: ivanivanov
```

Настройка прокси-авторизации

Содержит два основных блока:

- `headers` — маппинг заголовков в атрибуты пользователя.
- `allowCIDRs` — список IP/подсетей, с которых разрешён приём заголовков авторизации.

В блоке `headers` содержатся следующие поля:

| Название поля | Описание | Пример значения |
|------------------------|---|---------------------------------|
| <code>email</code> | Содержит email пользователя. | <code>email</code> |
| <code>username</code> | Обязательное поле — содержит идентификатор (имя) пользователя | <code>preferred_username</code> |
| <code>firstName</code> | Заголовок, содержащий имя | <code>given_name</code> |
| <code>lastName</code> | Заголовок, содержащий фамилию | <code>family_name</code> |
| <code>role</code> | Заголовок, содержащий роль пользователя. Если окажется пустым, то будет подставлена <code>defaultRole</code> (описана ниже) | <code>deployerAccess</code> |

Пример конфигурации:

```
auth:
  proxyAuth:
    headers:
      email: x-email
      username: x-username
      firstName: x-given-name
      lastName: x-family-name
      role: x-role
    allowCIDRs:
      - 172.20.70.190
```

Переопределение атрибутов пользователей

Состоит из имен пользователей, атрибуты которых необходимо переопределить. Например у нас есть пользователь `ivanivanov`:

| Название поля | Описание | Пример значения |
|---------------|--|--------------------|
| email | Содержит email пользователя. | ivanivanov@mail.ru |
| firstName | Заголовок, содержащий имя | ivan |
| lastName | Заголовок, содержащий фамилию | ivanov |
| role | Заголовок, содержащий роль пользователя. Если окажется пустым, то будет подставлена defaultRole (описана ниже) | admin |

Пример конфигурации:

```
auth:
  overrideUsers:
    as:
      email: ivanivanov@mail.ru
      firstName: ivan
      lastName: ivanov
      role: admin
```

Дополнительные настройки авторизации

Кроме четырех блоков описанных выше, есть еще несколько полей для настройки авторизации:

- `defaultRole` — Роль по умолчанию, на случай, если из внешней системы получена пустая роль.
- `auditLogPath` — Путь к audit-логу установщика. Если поле пустое, то audit-лог не будет записываться.
- `realIpHeader` — Заголовок с реальным IP клиента, если установщик работает за прокси-сервером.
- `realIpFrom` — Список IP/подсетей, с которых разрешён приём заголовка с реальным IP клиента. Если список пуст, то заголовок будет принят с любого адреса.

Пример конфигурации:

```
auth:
  ...
  defaultRole: admin
auditLogPath: audit.log
realIpHeader: x-real-IP
realIpFrom:
  - 172.20.70.190
```

Пример полной конфигурации

Скачать файл конфигурации

```
auth:
  oidc:
    host: https://biz.ws.vkwm1.on-premise.ru/auth/realms/EXCH.ON-PREMISE.RU/protocol/openid-
connect
    skipTls: false
    clientId: api
    clientSecret: otBrIQRxqvXj9VUt0k5iR1TcuBsmDzx8
    authPath: /auth
    tokenPath: /token
    infoPath: /userinfo
    scope: email
    disableRefresh: false
    infoFields:
      email: email
      username: preferred_username
      firstName: given_name
      lastName: family_name
      role: deployerAccess
  byIpUsers:
    - cidr: 172.20.70.190
      username: as
  proxyAuth:
    headers:
      email: x-email
      username: x-username
      firstName: x-given-name
      lastName: x-family-name
      role: x-role
    allowCIDRs:
      - 172.20.70.190
  overrideUsers:
    as:
      email: as@mail.ru
      firstName: a
      lastName: s
      role: admin
    defaultRole: admin
auditLogPath: audit.log
realIpHeader: x-real-IP
realIpFrom:
  - 172.20.70.190
```

Как получить логи авторизации через OIDC-провайдер

Информацию об авторизации можно увидеть в логах установщика:

```
journalctl -fu deployer
```

Либо на стороне OIDC-провайдера. Например, на сервере с Keycloak выполнить команду:

```
journalctl -fu onpremise-container-keycloak1
```

Как закрыть доступ до установщика с помощью TLS

Добавьте ключ и сертификат в строку запуска установщика:

```
./onpremise-deployer_linux -tlsCert fullchain.pem -tlsKey privkey.pem
```

Если установщик запускается через systemd:

1. Откройте для редактирования файл `/etc/systemd/system/deployer.service`:

```
vim /etc/systemd/system/deployer.service
```

2. В строчке `Exec` добавьте:

```
-tlsCert /etc/certs/fullchain.pem -tlsKey /etc/certs/privkey.pem
```

Указав полный путь до ключей.

3. Выполните команды:

```
systemctl daemon-reload  
systemctl restart deployer
```

 Автор: Груздев Никита

 30 июня 2025 г.