Администрирование сервера

Оглавление

Возможности и концепции Tuxedo SSO

Tuxedo SSO — это решение для единого входа в веб-приложения и веб-сервисы RESTful. Цель Tuxedo SSO — сделать безопасность простой, чтобы разработчикам приложений было легко защищать приложения и сервисы, которые они развернули в своей организации. Функции безопасности, которые разработчикам обычно приходится писать самостоятельно, предоставляются из коробки и легко настраиваются в соответствии с индивидуальными требованиями вашей организации. Tuxedo SSO предоставляет настраиваемые пользовательские интерфейсы для входа, регистрации, администрирования и управления учетными записями. Вы также можете использовать Tuxedo SSO в качестве платформы интеграции для подключения к существующим серверам LDAP и Active Directory. Вы также можете делегировать аутентификацию сторонним поставщикам удостоверений, таким как Facebook и Google.

Функции

Tuxedo SSO предоставляет следующие возможности:

- Единый вход и выход для браузерных приложений.
- Поддержка OpenID Connect.
- Поддержка OAuth 2.0.
- Поддержка SAML.
- Посредничество в идентификации — аутентификация с помощью внешних поставщиков идентификации OpenID Connect или SAML.
- Вход через социальные сети — включите вход через Google, GitHub, Facebook, Twitter и другие социальные сети.
- Федерация пользователей — синхронизация пользователей с серверами LDAP и Active Directory.
- Мост Kerberos — автоматическая аутентификация пользователей, вошедших на сервер Kerberos.
- Консоль администратора для централизованного управления пользователями, ролями, сопоставлениями ролей, клиентами и конфигурацией.
- Консоль учетной записи, позволяющая пользователям централизованно управлять своими учетными записями.
- Поддержка тем — настройте все страницы, доступные пользователю, для интеграции с вашими приложениями и брендингом.
- Двухфакторная аутентификация — поддержка TOTP/HOTP через Google Authenticator или FreeOTP.
- Процессы входа в систему — необязательная самостоятельная регистрация пользователя, восстановление пароля, проверка адреса электронной почты, запрос обновления пароля и т. д.

- Управление сеансами — администраторы и сами пользователи могут просматривать и управлять сеансами пользователей.
- Преобразователи токенов — преобразуйте атрибуты пользователя, роли и т. д. в токены и операторы нужным вам образом.
- Политики отзыва «не ранее» для каждой области, приложения и пользователя.
- Поддержка CORS. Клиентские адаптеры имеют встроенную поддержку CORS.
- Интерфейсы поставщика услуг (SPI) — ряд SPI, позволяющих настраивать различные аспекты сервера. Потоки аутентификации, поставщики федерации пользователей, сопоставители протоколов и многое другое.
- Поддерживает любую платформу/язык, имеющий библиотеку OpenID Connect Relying Party или библиотеку SAML 2.0 Service Provider.

Базовые операции Tuxedo SSO

Tuxedo SSO — это отдельный сервер, которым вы управляете в своей сети. Приложения настроены так, чтобы указывать на этот сервер и быть защищенными им. Tuxedo SSO использует открытые стандарты протоколов, такие как OpenID Connect или SAML 2.0, для защиты ваших приложений. Приложения браузера перенаправляют браузер пользователя из приложения на сервер аутентификации Tuxedo SSO, где они вводят свои учетные данные. Это перенаправление важно, поскольку пользователи полностью изолированы от приложений, и приложения никогда не видят учетные данные пользователя. Вместо этого приложениям предоставляется идентификационный токен или утверждение, которое криптографически подписано. Эти токены могут иметь идентификационную информацию, такую как имя пользователя, адрес, адрес электронной почты и другие данные профиля. Они также могут содержать данные разрешений, чтобы приложения могли принимать решения об авторизации. Эти токены также можно использовать для выполнения безопасных вызовов служб на основе REST.

Основные понятия и термины

Прежде чем пытаться использовать Tuxedo SSO для защиты ваших веб-приложений и служб REST, рассмотрите эти основные концепции и термины.

пользователи

: Пользователи — это сущности, которые могут входить в вашу систему. Они могут иметь атрибуты, связанные с ними, такие как адрес электронной почты, имя пользователя, адрес, номер телефона и день рождения. Им может быть назначено членство в группе и назначены определенные роли.

аутентификация

: Процесс идентификации и проверки пользователя.

авторизация

: Процесс предоставления доступа пользователю.

реквизиты для входа

: Учетные данные — это фрагменты данных, которые Tuxedo SSO использует для проверки личности пользователя. Некоторые примеры — пароли, одноразовые пароли, цифровые сертификаты или даже отпечатки пальцев.

роли

: Роли определяют тип или категорию пользователя. Admin, user, manager, и employeевсе это типичные роли, которые могут существовать в организации. Приложения часто назначают доступ и разрешения определенным ролям, а не отдельным пользователям, поскольку работа с пользователями может быть слишком детализированной и сложной для управления.

сопоставление ролей пользователей

: Сопоставление ролей пользователей определяет сопоставление между ролью и пользователем. Пользователь может быть связан с нулем или более ролей.

Эта информация о сопоставлении ролей может быть инкапсулирована в токены и утверждения, чтобы приложения могли определять разрешения на доступ к различным ресурсам, которыми они управляют.

составные роли

Составная роль — это роль, которая может быть связана с другими ролями. Например, superusercоставная роль может быть связана с ролями sales-adminи order-entry-admin. Если пользователь сопоставлен с superuserролью, он также наследует роли sales-adminи order-entry-admin.

группы

Группы управляют группами пользователей. Для группы можно определить атрибуты. Вы также можете сопоставлять роли с группой. Пользователи, которые становятся членами группы, наследуют атрибуты и сопоставления ролей, которые определяет группа.

сферы

Область управляет набором пользователей, учетных данных, ролей и групп. Пользователь принадлежит к области и входит в нее. Области изолированы друг от друга и могут управлять и аутентифицировать только тех пользователей, которых они контролируют.

клиенты

Клиенты — это сущности, которые могут запрашивать Tuxedo SSO для аутентификации пользователя. Чаще всего клиенты — это приложения и службы, которые хотят использовать Tuxedo SSO для своей защиты и предоставления решения для единого входа. Клиентами также могут быть сущности, которые просто хотят запросить идентификационную информацию или токен доступа, чтобы они могли безопасно вызывать другие службы в сети, защищенные Tuxedo SSO.

клиентские адаптеры

Клиентские адаптеры — это плагины, которые вы устанавливаете в среду своего приложения, чтобы иметь возможность общаться и быть защищенным

Tuxedo SSO. Tuxedo SSO имеет ряд адаптеров для разных платформ, которые вы можете загрузить. Существуют также сторонние адаптеры, которые вы можете получить для сред, которые мы не охватываем.

согласие

Согласие — это когда вы как администратор хотите, чтобы пользователь дал разрешение клиенту, прежде чем этот клиент сможет участвовать в процессе аутентификации. После того, как пользователь предоставит свои учетные данные, Tuxedo SSO выведет всплывающее окно, идентифицирующее клиента, запрашивающего вход, и какую идентификационную информацию запрашивают у пользователя. Пользователь может решить, предоставлять или нет запрос.

области действия клиента

Когда клиент зарегистрирован, вы должны определить сопоставители протоколов и сопоставления областей ролей для этого клиента. Часто бывает полезно хранить область клиента, чтобы упростить создание новых клиентов путем совместного использования некоторых общих настроек. Это также полезно для запроса некоторых утверждений или ролей, которые будут условно основаны на значении scopeпараметра. Tuxedo SSO предоставляет для этого концепцию области клиента.

роль клиента

Клиенты могут определять роли, которые являются специфическими для них. По сути, это пространство имен ролей, выделенное для клиента.

идентификационный токен

Токен, предоставляющий идентификационную информацию о пользователе. Часть спецификации OpenID Connect.

токен доступа

Токен, который может быть предоставлен как часть HTTP-запроса, предоставляющего доступ к вызываемой службе. Это часть спецификации OpenID Connect и OAuth 2.0.

утверждение

Информация о пользователе. Обычно это относится к XML-блоку, который включен в ответ аутентификации SAML, который предоставил метаданные идентификации об аутентифицированном пользователе.

учетная запись службы

У каждого клиента есть встроенная учетная запись сервиса, которая позволяет ему получить токен доступа.

прямой грант

Способ получения клиентом токена доступа от имени пользователя посредством вызова REST.

картографы протоколов

Для каждого клиента вы можете настроить, какие утверждения и заявления хранятся в токене OIDC или утверждении SAML. Вы делаете это для каждого клиента, создавая и настраивая сопоставители протоколов.

сессия

Когда пользователь входит в систему, создается сеанс для управления сеансом входа. Сеанс содержит информацию, например, когда пользователь вошел в систему и какие приложения участвовали в едином входе в течение этого сеанса. И администраторы, и пользователи могут просматривать информацию о сеансе.

поставщик федерации пользователей

Tuxedo SSO может хранить и управлять пользователями. Часто компании уже имеют службы LDAP или Active Directory, которые хранят информацию о пользователях и учетных данных. Вы можете указать Tuxedo SSO проверять учетные данные из этих внешних хранилищ и извлекать информацию об идентификации.

поставщик удостоверений личности

Поставщик удостоверений (IDP) — это сервис, который может аутентифицировать пользователя. Tuxedo SSO — это IDP.

федерация поставщиков удостоверений

Tuxedo SSO можно настроить для делегирования аутентификации одному или нескольким IDP. Социальный вход через Facebook или Google является примером федерации поставщиков удостоверений. Вы также можете подключить Tuxedo SSO для делегирования аутентификации любому другому OpenID Connect или SAML 2.0 IDP.

картографы поставщиков удостоверений

При выполнении федерации IDP вы можете сопоставить входящие токены и утверждения с атрибутами пользователя и сеанса. Это поможет вам распространить информацию об идентичности от внешнего IDP к вашему клиенту, запрашивающему аутентификацию.

необходимые действия

Обязательные действия — это действия, которые пользователь должен выполнить в процессе аутентификации. Пользователь не сможет завершить процесс аутентификации, пока эти действия не будут выполнены. Например, администратор может запланировать пользователям ежемесячный сброс паролей. update passwordДля всех этих пользователей будет установлено обязательное действие.

потоки аутентификации

Потоки аутентификации — это рабочие потоки, которые пользователь должен выполнить при взаимодействии с определенными аспектами системы. Поток входа может определять, какие типы учетных данных требуются. Поток регистрации определяет, какую информацию профиля должен ввести пользователь и следует ли использовать что-то вроде reCAPTCHA для фильтрации ботов. Поток сброса учетных данных определяет, какие действия должен выполнить пользователь, прежде чем он сможет сбросить свой пароль.

события

События — это потоки аудита, которые администраторы могут просматривать и подключать.

темы

Каждый экран, предоставляемый Tuxedo SSO, поддерживается темой. Темы определяют HTML-шаблоны и таблицы стилей, которые вы можете переопределять по мере необходимости.

Создание первого администратора

После установки Tuxedo SSO вам понадобится учетная запись администратора, которая может выступать в качестве суперадминистратора с полными правами на управление Tuxedo SSO. С этой учетной записью вы можете войти в консоль администратора Tuxedo SSO, где вы создаете области и пользователей, а также регистрируете приложения, защищенные Tuxedo SSO.

Создание учетной записи на локальном хосте

Если ваш сервер доступен из localhost, выполните следующие действия.

Процедура

1. В веб-браузере перейдите по адресу http://localhost:8080 .
2. Введите имя пользователя и пароль, которые вы можете вспомнить.

   Приветственная страница

Создание учетной записи удаленно

Если вы не можете получить доступ к серверу по localhostадресу или просто хотите запустить Tuxedo SSO из командной строки, используйте переменные среды KC_BOOTSTRAP_ADMIN_USERNAMEи KC_BOOTSTRAP_ADMIN_PASSWORDдля создания начальной учетной записи администратора.

Например:

```
export KC_BOOTSTRAP_ADMIN_USERNAME=<username>
export KC_BOOTSTRAP_ADMIN_PASSWORD=<password>
```

bin/kc.[sh|bat] start

Настройка областей

После того, как у вас есть административная учетная запись для консоли администратора, вы можете настроить области. Область — это пространство, в котором вы управляете объектами, включая пользователей, приложения, роли и группы. Пользователь принадлежит к области и входит в нее. Одно развертывание Tuxedo SSO может определять, хранить и управлять таким количеством областей, для которого есть место в базе данных.

Использование консоли администратора

Вы можете настраивать области и выполнять большинство административных задач в консоли администратора Tuxedo SSO.

Предпосылки

- Вам нужна учетная запись администратора. Смотрите Создание первого администратора .

Процедура

1. Перейдите по URL-адресу консоли администратора.

   Например, для localhost используйте этот URL: http://localhost:8080/admin/

   Страница входа

2. Введите имя пользователя и пароль, которые вы создали на странице приветствия или через переменные среды, как указано в руководстве Создание начального администратора . Это действие отображает консоль администратора.

   Консоль администратора

3. Обратите внимание на меню и другие опции, которые вы можете использовать:
    - Нажмите меню « Мастер» , чтобы выбрать область, которой вы хотите управлять, или создать новую.
    - Нажмите на верхний правый список, чтобы просмотреть свою учетную запись или выйти из системы.
    - Наведите курсор на значок вопросительного знака ? , чтобы увидеть текст подсказки, описывающий это поле. Изображение выше показывает подсказку в действии.
    - Нажмите на значок вопросительного знака ? , чтобы отобразить текст подсказки, описывающий это поле. Изображение выше показывает подсказку в действии.

Экспорт файлов из консоли администратора не подходит для резервного копирования или передачи данных между серверами. Для резервного копирования или передачи данных между серверами подходит только экспорт во время загрузки.

Главное царство

В консоли администратора существует два типа областей:

- Master realm- Эта область была создана для вас, когда вы впервые запустили Tuxedo SSO. Она содержит учетную запись администратора, которую вы создали при первом входе в систему. Используйте главную область только для создания и управления областями в вашей системе.
- Other realms- Эти области создаются администратором в главной области. В этих областях администраторы управляют пользователями в вашей организации и необходимыми им приложениями. Приложения принадлежат пользователям.

Области и приложения

Области изолированы друг от друга и могут управлять и аутентифицировать только тех пользователей, которых они контролируют. Следование этой модели безопасности помогает предотвратить случайные изменения и следует традиции

предоставления учетным записям пользователей доступа только к тем привилегиям и полномочиям, которые необходимы для успешного выполнения их текущей задачи.

Дополнительные ресурсы

- См. Выделенные консоли администратора области , если вы хотите отключить главную область и определить учетные записи администратора в любой новой области, которую вы создаете. Каждая область имеет свою собственную выделенную консоль администратора, в которую вы можете войти с помощью локальных учетных записей.

Создание области

Вы создаете область, чтобы предоставить пространство управления, где вы можете создавать пользователей и давать им разрешения на использование приложений. При первом входе в систему вы обычно находитесь в главной области, области верхнего уровня, из которой вы создаете другие области.

Принимая решение о том, какие области вам нужны, подумайте о том, какой тип изоляции вы хотите иметь для своих пользователей и приложений. Например, вы можете создать область для сотрудников вашей компании и отдельную область для клиентов. Ваши сотрудники будут входить в область сотрудников и смогут посещать только внутренние приложения компании. Клиенты будут входить в область клиентов и смогут взаимодействовать только с приложениями, ориентированными на клиентов.

Процедура

1. Нажмите Tuxedo SSO рядом с главной областью , затем нажмите Создать область .

   Добавить меню области

2. Введите имя области.
3. Нажмите «Создать» .

   Создать область

Текущая область теперь установлена на область, которую вы только что создали. Вы можете переключаться между областями, нажимая на имя области в меню.

Настройка SSL для области

Каждая область имеет связанный режим SSL, который определяет требования SSL/HTTPS для взаимодействия с областью. Браузеры и приложения, которые взаимодействуют с областью, соблюдают требования SSL/HTTPS, определенные режимом SSL, иначе они не смогут взаимодействовать с сервером.

Процедура

1. Нажмите «Настройки области» в меню.
2. Перейдите на вкладку «Общие» .

   Вкладка «Общие»
3. Установите для параметра «Требовать SSL» один из следующих режимов SSL:
   - Внешние запросы Пользователи могут взаимодействовать с Tuxedo SSO без SSL, пока они придерживаются частных адресов IPv4, таких как localhost, 127.0.0.1, 10.x.x.x, 192.168.x.x, 172.16.x.хили IPv6-адресов link-local и unique-local. Если вы попытаетесь получить доступ к Tuxedo SSO без SSL с нечастного IP-адреса, вы получите ошибку.
   - None Tuxedo SSO не требует SSL. Этот выбор применим только в разработке, когда вы экспериментируете и не планируете поддерживать это развертывание.
   - Все запросы Tuxedo SSO требуют SSL для всех IP-адресов.

Настройка электронной почты для области

Tuxedo SSO отправляет электронные письма пользователям для проверки их адресов электронной почты, когда они забывают свои пароли или когда

администратору необходимо получать уведомления о событиях сервера. Чтобы разрешить Tuxedo SSO отправлять электронные письма, вы предоставляете Tuxedo SSO настройки вашего SMTP-сервера.

Процедура

1. Нажмите «Настройки области» в меню.
2. Нажмите вкладку «Электронная почта» .

   Вкладка «Электронная почта»
3. Заполните поля и переключите переключатели по мере необходимости.

Шаблон

От

> From обозначает адрес, используемый для заголовка SMTP From отправляемых писем.

Из отображаемого имени

> Отображаемое имя From позволяет настроить удобные псевдонимы адреса электронной почты (необязательно). Если не задано, в почтовых клиентах будет отображаться простой адрес From .

Отвечать на

> Reply to обозначает адрес, используемый для заголовка SMTP Reply-To для отправленных писем (необязательно). Если не задано, будет использоваться простой адрес электронной почты From .

Ответить на отображаемое имя

> Reply to display name позволяет настроить удобные для пользователя псевдонимы адреса электронной почты (необязательно). Если не задано, будет отображаться простой адрес электронной почты Reply To .

Конверт от

Конверт from обозначает адрес возврата , используемый для заголовка SMTP Return-Path для отправленных писем (необязательно).

Подключение и аутентификация

Хозяин

Хост обозначает имя хоста SMTP-сервера, используемого для отправки электронных писем.

Порт

Порт обозначает порт SMTP-сервера.

Шифрование

Отметьте один из этих флажков, чтобы поддерживать отправку писем для восстановления имен пользователей и паролей, особенно если SMTP-сервер находится во внешней сети. Скорее всего, вам потребуется изменить порт на 465, порт по умолчанию для SSL/TLS.

Аутентификация

Установите этот переключатель в положение ON, если ваш SMTP-сервер требует аутентификации. При появлении запроса укажите Имя пользователя и Пароль . Значение поля Пароль может ссылаться на значение из внешнего хранилища .

Настройка тем

Для определенной области вы можете изменить внешний вид любого пользовательского интерфейса в Tuxedo SSO с помощью тем.

Процедура

1. Нажмите «Настройка области» в меню.
2. Нажмите вкладку Темы .

   Вкладка «Темы»

3. Выберите нужную тему для каждой категории пользовательского интерфейса и нажмите «Сохранить» .

   Тема входа

   Ввод имени пользователя и пароля, ввод одноразового пароля, регистрация нового пользователя и другие подобные экраны, связанные с входом в систему.

   Тема аккаунта

   Консоль, используемая пользователем для управления своей учетной записью.

   Тема консоли администратора

   Скин консоли администратора Tuxedo SSO.

   Тема электронной почты

   Всякий раз, когда Tuxedo SSO отправляет электронное письмо, он использует шаблоны, определенные в этой теме, для его создания.

Дополнительные ресурсы

- В руководстве разработчика сервера описывается, как создать новую тему или изменить существующие.

Обеспечение интернационализации

Каждый экран пользовательского интерфейса интернационализирован в Tuxedo SSO. Язык по умолчанию — английский, но вы можете выбрать, какие локали вы хотите поддерживать и какой будет локаль по умолчанию.

Процедура

1. Нажмите «Настройки области» в меню.
2. Нажмите вкладку Локализация .

3. Включить интернационализацию .
4. Выберите языки, которые вы будете поддерживать.

   Вкладка «Локализация»

   При следующем входе пользователя в систему он сможет выбрать язык на странице входа, который будет использоваться для экранов входа, консоли учетной записи и консоли администратора.

Дополнительные ресурсы

- В руководстве разработчика сервера объясняется, как можно предложить дополнительные языки. Все интернационализированные тексты, предоставляемые темой, могут быть перезаписаны текстами, специфичными для области, на вкладке Локализация .

Выбор локали пользователя

Поставщик выбора локали предлагает лучшую локаль на основе доступной информации. Однако часто неизвестно, кто является пользователем. По этой причине локаль ранее аутентифицированного пользователя запоминается в сохраняемом файле cookie.

Логика выбора локали использует первый из следующих доступных вариантов:

- Выбрано пользователем — когда пользователь выбрал локаль с помощью раскрывающегося списка выбора локали.
- Профиль пользователя — когда есть аутентифицированный пользователь и у пользователя установлена предпочтительная локаль.
- Выбрано клиентом — передается клиентом, например, с помощью параметра ui_locales
- Файл cookie — последняя выбранная в браузере локаль
- Принятый язык - локаль из заголовка Accept-Language
- Область по умолчанию

- Если ничего из вышеперечисленного не подходит, вернитесь к английскому языку.

Когда пользователь проходит аутентификацию, запускается действие по обновлению локали в сохраненном файле cookie, упомянутом ранее. Если пользователь активно переключил локаль с помощью селектора локали на страницах входа, локаль пользователя также обновляется в этот момент.

Если вы хотите изменить логику выбора локали, у вас есть возможность создать пользовательский LocaleSelectorProvider. Для получения подробной информации обратитесь к Руководству разработчика сервера .

Управление параметрами входа

Tuxedo SSO включает в себя несколько встроенных функций страницы входа.

Включение функции «Забыли пароль»

Если вы включите Forgot password, пользователи смогут сбросить свои учетные данные для входа, если они забудут свои пароли или потеряют генератор одноразовых паролей.

Процедура

1. Нажмите «Настройки области» в меню.
2. Нажмите вкладку «Вход» .

   Вкладка «Вход»
3. Переключите опцию «Забыли пароль» в положение ВКЛ .

   Ссылка Forgot Password?отображается на страницах входа.

   Забыли пароль? Ссылка
4. Укажите Hostи Fromна вкладке «Электронная почта» , чтобы Tuxedo SSO мог отправить электронное письмо для сброса пароля.

5. Щелкните эту ссылку, чтобы перенаправить пользователей на страницу, где они могут ввести свое имя пользователя или адрес электронной почты и получить электронное письмо со ссылкой для сброса своих учетных данных.

   Забыли пароль?

Текст, отправляемый в электронном письме, можно настроить. Для получения дополнительной информации см. Руководство разработчика сервера .

Когда пользователи нажимают на ссылку электронной почты, Tuxedo SSO просит их обновить пароль, а если они настроили генератор OTP, Tuxedo SSO просит их перенастроить генератор OTP. В зависимости от требований безопасности вашей организации вы можете не захотеть, чтобы пользователи сбрасывали свой генератор OTP по электронной почте.

Чтобы изменить это поведение, выполните следующие действия:

Процедура

1. Нажмите «Аутентификация» в меню.
2. Перейдите на вкладку Потоки .
3. Выберите процесс сброса учетных данных .

   Сброс потока учетных данных

   Если вы не хотите сбрасывать OTP, установите Reset - Conditional OTPдля требования подпотока значение Отключено .
4. Нажмите «Аутентификация» в меню.
5. Нажмите вкладку «Требуемые действия» .
6. Убедитесь, что функция обновления пароля включена.

   Требуемые действия

Включение функции «Запомнить меня»

Пользователь, вошедший в систему, закрывая свой браузер, уничтожает свою сессию, и этот пользователь должен войти в систему снова. Вы можете настроить

Tuxedo SSO так, чтобы сессия входа пользователя оставалась открытой, если этот пользователь щелкает флажок «Запомнить меня» при входе в систему. Это действие превращает cookie входа из cookie только для сессии в cookie сохранения.

Процедура

1. Нажмите «Настройки области» в меню.
2. Нажмите вкладку «Вход» .
3. Установите переключатель «Запомнить меня» в положение «Вкл.» .

   Вкладка «Вход»

   При сохранении этой настройки remember мена странице входа в область появится флажок.

   Запомнить меня

Сопоставление ACR с уровнем аутентификации (LoA)

В общих настройках области можно определить, какое Authentication Context Class Reference (ACR)значение сопоставляется с каким Level of Authentication (LoA). ACR может быть любым значением, тогда как LoA должен быть числовым. Утверждение acr может быть запрошено в параметре claimsили acr_values, отправленном в запросе OIDC, и оно также включено в токен доступа и токен ID. Сопоставленный номер используется в условиях потока аутентификации.

Сопоставление может быть также указано на уровне клиента в случае, если конкретному клиенту необходимо использовать значения, отличные от realm. Однако лучшей практикой является придерживаться сопоставлений realm.

Более подробную информацию см. в разделе «Пошаговая аутентификация» и официальной спецификации OIDC .

Обновление рабочего процесса электронной почты (UpdateEmail)

При использовании этого рабочего процесса пользователям придется использовать действие UPDATE_EMAIL, чтобы изменить свой адрес электронной почты.

Действие связано с одной формой ввода адреса электронной почты. Если в области отключена проверка адреса электронной почты, это действие позволит обновить адрес электронной почты без проверки. Если в области включена проверка адреса электронной почты, действие отправит токен действия обновления адреса электронной почты на новый адрес электронной почты без изменения адреса электронной почты учетной записи. Только срабатывание токена действия завершит обновление адреса электронной почты.

Приложения могут отправлять своих пользователей на форму обновления электронной почты, используя UPDATE_EMAIL в качестве AIA (Application Initiated Action) .

> UpdateEmail — это **предварительная версия** , которая не поддерживается полностью. Эта функция отключена по умолчанию.
>
> Чтобы включить запустите сервер с помощью `--features=preview`или`--features=update-email`
>
> Если вы включаете эту функцию и переходите с предыдущей версии, включите действие **Update Email** required в своих областях. В противном случае пользователи не смогут обновить свои адреса электронной почты.

Настройка ключей области

Протоколы аутентификации, которые использует Tuxedo SSO, требуют криптографических подписей и иногда шифрования. Для этого Tuxedo SSO использует асимметричные пары ключей, закрытый и открытый ключ.

Tuxedo SSO имеет одну активную пару ключей одновременно, но может иметь и несколько пассивных ключей. Активная пара ключей используется для создания новых подписей, в то время как пассивная пара ключей может использоваться для проверки предыдущих подписей. Это позволяет регулярно менять ключи без простоев или помех для пользователей.

При создании области автоматически генерируется пара ключей и самоподписанный сертификат.

Процедура

1. Нажмите «Настройки области» в меню.
2. Нажмите «Ключи» .
3. Для просмотра пассивных ключей выберите «Пассивные ключи» в раскрывающемся списке фильтров.
4. Для просмотра отключенных ключей выберите «Отключенные ключи» в раскрывающемся списке фильтров.

Пара ключей может иметь статус Active, но все еще не быть выбранной в качестве текущей активной пары ключей для области. Выбранная активная пара, которая используется для подписей, выбирается на основе первого поставщика ключей, отсортированного по приоритету, который может предоставить активную пару ключей.

Вращающиеся ключи

Мы рекомендуем вам регулярно менять ключи. Начните с создания новых ключей с более высоким приоритетом, чем у существующих активных ключей. Вместо этого вы можете создать новые ключи с тем же приоритетом и сделать предыдущие ключи пассивными.

Как только новые ключи станут доступны, все новые токены и куки будут подписаны новыми ключами. Когда пользователь проходит аутентификацию в приложении, куки SSO обновляется новой подписью. Когда токены OpenID Connect обновляются, новые токены подписываются новыми ключами. В конце концов, все куки и токены используют новые ключи, и через некоторое время старые ключи можно удалить.

Частота удаления старых ключей — это компромисс между безопасностью и необходимостью обновления всех файлов cookie и токенов. Рассмотрите возможность создания новых ключей каждые три-шесть месяцев и удаления старых ключей через один-два месяца после создания новых ключей. Если

пользователь был неактивен в период между добавлением новых ключей и удалением старых ключей, этому пользователю придется пройти повторную аутентификацию.

Ротация ключей также применима к офлайн-токенам. Чтобы убедиться, что они обновлены, приложениям необходимо обновить токены до того, как старые ключи будут удалены.

Добавление сгенерированной пары ключей

Используйте эту процедуру для создания пары ключей, включая самоподписанный сертификат.

Процедура

1. Выберите область в консоли администратора.
2. Нажмите «Настройки области» в меню.
3. Нажмите вкладку «Ключи» .
4. Перейдите на вкладку «Поставщики» .
5. Нажмите «Добавить поставщика» и выберите файл rsa-generated .
6. Введите число в поле Приоритет . Это число определяет, станет ли новая пара ключей активной парой ключей. Наибольшее число делает пару ключей активной.
7. Выберите значение размера ключа AES .
8. Нажмите «Сохранить ».

Изменение приоритета поставщика не приведет к повторной генерации ключей, но если вы хотите изменить размер ключа, вы можете отредактировать поставщика, и будут сгенерированы новые ключи.

Ротация ключей путем извлечения сертификата

Вы можете выполнить ротацию ключей, извлекая сертификат из пары ключей, сгенерированной RSA, и используя этот сертификат в новом хранилище ключей.

Предпосылки

- Сгенерированная пара ключей

Процедура

1. Выберите область в консоли администратора.
2. Нажмите Настройки области .
3. Нажмите вкладку «Ключи» .

   Появится список активных ключей.
4. В строке с ключом RSA нажмите Сертификат в разделе Открытые ключи .

   Сертификат представлен в текстовой форме.
5. Сохраните сертификат в файле и вложите его в эти строки.

   ----Begin Certificate----
   <Output>
   ----End Certificate----
6. Используйте команду keytool для преобразования файла ключа в формат PEM.
7. Удалите текущий сертификат открытого ключа RSA из хранилища ключей.

   keytool -delete -keystore <keystore>.jks -storepass <password> -alias <key>
8. Импортируйте новый сертификат в хранилище ключей.

   keytool -importcert -file domain.crt -keystore <keystore>.jks -storepass <password> -alias <key>
9. Перестройте приложение.

   mvn clean install wildfly:deploy

Добавление существующей пары ключей и сертификата

Чтобы добавить пару ключей и сертификат, полученный в другом месте, выберите Providersи выберите rsaиз раскрывающегося списка. Вы можете изменить приоритет, чтобы убедиться, что новая пара ключей станет активной парой ключей.

Предпосылки

- Файл закрытого ключа. Файл должен быть в формате PEM.

Процедура

1. Выберите область в консоли администратора.
2. Нажмите Настройки области .
3. Нажмите вкладку «Ключи» .
4. Перейдите на вкладку «Поставщики» .
5. Нажмите Добавить провайдера и выберите rsa .
6. Введите число в поле Приоритет . Это число определяет, станет ли новая пара ключей активной парой ключей.
7. Нажмите кнопку «Обзор…» рядом с «Закрытый ключ RSA», чтобы загрузить файл закрытого ключа.
8. Если у вас есть подписанный сертификат для вашего закрытого ключа, нажмите Browse… рядом с X509 Certificate , чтобы загрузить файл сертификата. Tuxedo SSO автоматически генерирует самоподписанный сертификат, если вы не загрузите сертификат.
9. Нажмите «Сохранить ».

Загрузка ключей из хранилища ключей Java

Чтобы добавить пару ключей и сертификат, хранящиеся в файле Java Keystore на хосте, выберите Providersи выберите java-keystoreиз раскрывающегося списка. Вы можете изменить приоритет, чтобы убедиться, что новая пара ключей станет активной парой ключей.

Для загрузки соответствующей цепочки сертификатов ее необходимо импортировать в файл хранилища ключей Java, который Key Aliasиспользовался для загрузки пары ключей.

Процедура

1. Выберите область в консоли администратора.

2. Нажмите «Настройки области» в меню.
3. Нажмите вкладку «Ключи» .
4. Перейдите на вкладку «Поставщики» .
5. Нажмите Добавить поставщика и выберите java-keystore .
6. Введите число в поле Приоритет . Это число определяет, станет ли новая пара ключей активной парой ключей.
7. Введите нужный алгоритм . Обратите внимание, что алгоритм должен соответствовать типу ключа (например, RS256требуется закрытый ключ RSA, ES256закрытый ключ EC или AESсекретный ключ AES).
8. Введите значение для Keystore . Путь к файлу хранилища ключей.
9. Введите пароль хранилища ключей . Параметр может ссылаться на значение из внешнего хранилища .
10. Введите значение для типа хранилища ключей (JKS, PKCS12или BCFKS).
11. Введите значение псевдонима ключа, который будет загружен из хранилища ключей.
12. Введите пароль ключа . Опция может ссылаться на значение из внешнего хранилища .
13. Введите значение для Key Use (sigдля подписи или encдля шифрования). Обратите внимание, что использование должно соответствовать типу алгоритма (например, RS256is, signo RSA-OAEPis enc)
14. Нажмите «Сохранить ».

Не все типы хранилищ ключей поддерживают все типы ключей. Например, `JKS`во всех режимах и `PKCS12`в режиме fips (`BCFIPS`поставщик) не может хранить записи секретных ключей.

Делаем ключи пассивными

Процедура

1. Выберите область в консоли администратора.

2. Нажмите «Настройки области» в меню.
3. Нажмите вкладку «Ключи» .
4. Перейдите на вкладку «Поставщики» .
5. Щелкните поставщика ключа, который вы хотите сделать пассивным.
6. Переключите «Активно» в положение «Выкл.» .
7. Нажмите «Сохранить ».

Отключение клавиш

Процедура

1. Выберите область в консоли администратора.
2. Нажмите «Настройки области» в меню.
3. Нажмите вкладку «Ключи» .
4. Перейдите на вкладку «Поставщики» .
5. Щелкните поставщика ключа, который вы хотите сделать пассивным.
6. Переключите «Включено» в «Выкл .» .
7. Нажмите «Сохранить ».

Скомпрометированные ключи

Tuxedo SSO хранит ключи подписи только локально и никогда не делится ими с клиентскими приложениями, пользователями или другими субъектами. Однако если вы считаете, что ваш ключ подписи области был скомпрометирован, вам следует сначала сгенерировать новую пару ключей, как описано выше, а затем немедленно удалить скомпрометированную пару ключей.

Либо вы можете удалить поставщика из Providerстаблицы.

Процедура

1. Нажмите «Клиенты» в меню.
2. Нажмите security-admin-console .

3. Прокрутите страницу вниз до раздела «Настройки доступа» .
4. Заполните поле URL-адрес администратора .
5. Нажмите вкладку «Дополнительно» .
6. Нажмите Установить сейчас в разделе Отзыв .
7. Нажмите «Отправить» .

Применение политики not-before гарантирует, что клиентские приложения не примут существующие токены, подписанные скомпрометированным ключом. Клиентское приложение также вынуждено загрузить новые пары ключей из Tuxedo SSO, поэтому токены, подписанные скомпрометированным ключом, будут недействительными.

Клиенты REST и конфиденциальные клиенты должны задать **URL-адрес администратора** , чтобы Tuxedo SSO мог отправлять клиентам запрос политики Not-Pre.

Использование внешнего хранилища

Организации могут иметь базы данных, содержащие информацию, пароли и другие учетные данные. Обычно вы не можете перенести существующее хранилище данных в развертывание Tuxedo SSO, поэтому Tuxedo SSO может объединить существующие внешние базы данных пользователей. Tuxedo SSO поддерживает LDAP и Active Directory, но вы также можете кодировать расширения для любой пользовательской базы данных с помощью Tuxedo SSO User Storage SPI.

Когда пользователь пытается войти в систему, Tuxedo SSO проверяет хранилище этого пользователя, чтобы найти этого пользователя. Если Tuxedo SSO не находит пользователя, Tuxedo SSO выполняет итерацию по каждому поставщику хранилища пользователя для области, пока не найдет совпадение. Затем данные из внешнего хранилища данных сопоставляются со стандартной моделью пользователя, которую использует среда выполнения Tuxedo SSO. Затем эта модель пользователя сопоставляется с утверждениями токенов OIDC и атрибутами утверждений SAML.

Внешние базы данных пользователей редко содержат данные, необходимые для поддержки всех функций Tuxedo SSO, поэтому поставщик хранилища пользователей может выбрать локальное хранение элементов в хранилище данных пользователей Tuxedo SSO. Поставщики могут импортировать пользователей локально и периодически синхронизировать с внешним хранилищем данных. Этот подход зависит от возможностей поставщика и его конфигурации. Например, ваше внешнее хранилище данных пользователей может не поддерживать OTP. OTP может обрабатываться и храниться Tuxedo SSO в зависимости от поставщика.

Добавление провайдера

Чтобы добавить поставщика хранилища, выполните следующую процедуру:

Процедура

1. Нажмите «Федерация пользователей» в меню.

   Федерация пользователей

2. Выберите тип карты провайдера из списка карт.

   Tuxedo SSO перенаправит вас на страницу конфигурации этого провайдера.

Решение проблем с поставщиками

Если поставщик хранилища пользователя выходит из строя, вы не сможете войти в систему и просматривать пользователей в консоли администратора. Tuxedo SSO не обнаруживает сбои при использовании поставщика хранилища для поиска пользователя, поэтому он отменяет вызов. Если у вас есть поставщик хранилища с высоким приоритетом, который выходит из строя во время поиска пользователя, вход в систему или запрос пользователя завершается с ошибкой и не будет переключен на следующего настроенного поставщика.

Tuxedo SSO сначала ищет в локальной базе данных пользователей Tuxedo SSO, чтобы разрешить пользователей до любого LDAP или пользовательского поставщика хранилища пользователей. Рассмотрите возможность создания учетной записи администратора, хранящейся в локальной базе данных пользователей Tuxedo SSO, на случай проблем с подключением к LDAP и бэкендам.

У каждого LDAP и пользовательского поставщика хранилища пользователей есть enableпереключатель на странице консоли администратора. Отключение поставщика хранилища пользователей пропускает поставщика при выполнении запросов, поэтому вы можете просматривать и входить в систему с учетными записями пользователей другого поставщика с более низким приоритетом. Если ваш поставщик использует importстратегию и отключен, импортированные пользователи по-прежнему доступны для поиска в режиме только для чтения.

Когда поиск поставщика хранилища завершается неудачей, Tuxedo SSO не выполняет отказ, поскольку в базах данных пользователей часто встречаются дублирующиеся имена пользователей или дублирующиеся адреса электронной почты. Дублирующиеся имена пользователей и адреса электронной почты могут вызывать проблемы, поскольку пользователь загружает данные из одного внешнего хранилища, когда администратор ожидает, что они будут загружаться из другого хранилища данных.

Облегченный протокол доступа к каталогам (LDAP) и Active Directory

Tuxedo SSO включает провайдера LDAP/AD. Вы можете объединить несколько различных серверов LDAP в одну область Tuxedo SSO и сопоставить атрибуты пользователя LDAP с общей моделью пользователя Tuxedo SSO.

По умолчанию Tuxedo SSO сопоставляет имя пользователя, адрес электронной почты, имя и фамилию учетной записи пользователя, но вы также можете настроить дополнительные сопоставления . Поставщик LDAP/AD Tuxedo SSO поддерживает проверку пароля с использованием протоколов LDAP/AD и режимов хранения, редактирования и синхронизации.

Настройка федеративного хранилища LDAP

Процедура

1. Нажмите «Федерация пользователей» в меню.

   Федерация пользователей

2. Нажмите Добавить поставщиков LDAP .

Tuxedo SSO перенесет вас на страницу конфигурации LDAP.

Режим хранения

Tuxedo SSO импортирует пользователей из LDAP в локальную базу данных пользователей Tuxedo SSO. Эта копия базы данных пользователей синхронизируется по требованию или через периодическую фоновую задачу. Исключение существует для синхронизации паролей. Tuxedo SSO никогда не импортирует пароли. Проверка пароля всегда происходит на сервере LDAP.

Преимущество синхронизации в том, что все функции Tuxedo SSO работают эффективно, поскольку все необходимые дополнительные данные для каждого пользователя хранятся локально. Недостаток в том, что каждый раз, когда Tuxedo SSO впервые запрашивает определенного пользователя, Tuxedo SSO выполняет соответствующую вставку в базу данных.

Вы можете синхронизировать импорт с вашим сервером LDAP. Синхронизация импорта не нужна, когда преобразователи LDAP всегда считывают определенные атрибуты из LDAP, а не из базы данных.

Вы можете использовать LDAP с Tuxedo SSO без импорта пользователей в базу данных пользователей Tuxedo SSO. Сервер LDAP создает резервную копию общей модели пользователя, которую использует среда выполнения Tuxedo SSO. Если LDAP не поддерживает данные, необходимые для функции Tuxedo SSO, эта функция не будет работать. Преимущество этого подхода в том, что у вас нет ресурсов для импорта и синхронизации копий пользователей LDAP в базу данных пользователей Tuxedo SSO.

Переключатель Import Users на странице конфигурации LDAP управляет этим режимом хранения. Чтобы импортировать пользователей, переключите этот переключатель в положение ON .

Если вы отключите **Import Users** , вы не сможете сохранять атрибуты профиля пользователя в базе данных Tuxedo SSO. Кроме того, вы не сможете сохранять метаданные, за исключением метаданных профиля пользователя, сопоставленных с LDAP. Эти метаданные могут включать сопоставления ролей, сопоставления групп и другие метаданные на основе конфигурации сопоставителей LDAP.

При попытке изменить данные пользователя, сопоставленные не с LDAP, обновление пользователя невозможно. Например, вы не можете отключить сопоставленного пользователя LDAP, если `enabled`флаг пользователя не сопоставлен с атрибутом LDAP.

Режим редактирования

Пользователи и администраторы могут изменять метаданные пользователя, пользователи через Account Console , а администраторы через Admin Console. Edit ModeКонфигурация на странице конфигурации LDAP определяет привилегии обновления LDAP пользователя.

ТОЛЬКО ДЛЯ ЧТЕНИЯ

> Вы не можете изменить имя пользователя, email, имя, фамилию и другие сопоставленные атрибуты. Tuxedo SSO показывает ошибку каждый раз, когда пользователь пытается обновить эти поля. Обновления паролей не поддерживаются.

ЗАПИСЫВАЕМЫЙ

> Вы можете изменить имя пользователя, адрес электронной почты, имя, фамилию и другие сопоставленные атрибуты и пароли и автоматически синхронизировать их с хранилищем LDAP.

НЕСИНХРОНИЗИРОВАНО

> Tuxedo SSO сохраняет изменения имени пользователя, электронной почты, имени, фамилии и паролей в локальном хранилище Tuxedo SSO, поэтому администратор должен синхронизировать эти данные обратно с LDAP. В этом режиме развертывания Tuxedo SSO могут обновлять метаданные пользователя на серверах LDAP только для чтения. Эта опция также применяется при импорте пользователей из LDAP в локальную базу данных пользователей Tuxedo SSO.

Когда Tuxedo SSO создает поставщика LDAP, Tuxedo SSO также создает набор начальных картографов LDAP . Tuxedo SSO настраивает эти карты на основе комбинации переключателей **Vendor** , **Edit Mode** и **Import Users** . Например, когда режим редактирования UNSYNCED, Tuxedo SSO настраивает карты для чтения определенного атрибута пользователя из базы данных, а не с сервера LDAP. Однако если вы позже измените режим редактирования, конфигурация карты не изменится, поскольку невозможно определить, были ли изменения

конфигурации изменены в режиме UNSYNCED. Определите режим редактирования **при** создании поставщика LDAP. Это примечание также применимо к переключателю **Import Users .**

Другие варианты конфигурации

Отображаемое имя консоли

Имя провайдера, которое будет отображаться в консоли администратора.

Приоритет

Приоритет провайдера при поиске пользователей или добавлении пользователя.

Синхронизация регистраций

Установите этот переключатель в положение ВКЛ, если вы хотите, чтобы новые пользователи, созданные Tuxedo SSO, добавлялись в LDAP.

Разрешить аутентификацию Kerberos

Включите аутентификацию Kerberos/SPNEGO в области с пользовательскими данными, предоставленными из LDAP. Для получения дополнительной информации см. раздел Kerberos .

Другие варианты

Наведите указатель мыши на подсказки в консоли администратора, чтобы увидеть более подробную информацию об этих параметрах.

Подключение к LDAP через SSL

При настройке URL-адреса безопасного соединения с вашим хранилищем LDAP (например, ldaps://myhost.com:636), Tuxedo SSO использует SSL для связи с сервером LDAP. Настройте хранилище доверия на стороне сервера Tuxedo SSO, чтобы Tuxedo SSO мог доверять соединению SSL с LDAP - см. руководство по настройке хранилища доверия .

Свойство Use Truststore SPIконфигурации устарело. Обычно его следует оставлять как Always.

Синхронизация пользователей LDAP с Tuxedo SSO

Если вы установите опцию Import Users , поставщик LDAP обрабатывает импорт пользователей LDAP в локальную базу данных Tuxedo SSO. Когда пользователь впервые входит в систему или возвращается как часть запроса пользователя (например, с помощью поля поиска в консоли администратора), поставщик LDAP импортирует пользователя LDAP в базу данных Tuxedo SSO. Во время аутентификации проверяется пароль LDAP.

Если вы хотите синхронизировать всех пользователей LDAP с базой данных Tuxedo SSO, настройте и включите параметры синхронизации на странице конфигурации поставщика LDAP.

Существуют два типа синхронизации:

Периодическая полная синхронизация

> Этот тип синхронизирует всех пользователей LDAP в базу данных Tuxedo SSO. Пользователи LDAP, которые уже есть в Tuxedo SSO, но отличаются в LDAP, напрямую обновляются в базе данных Tuxedo SSO.

Периодическая синхронизация измененных пользователей

> При синхронизации Tuxedo SSO создает или обновляет только пользователей, созданных или обновленных после последней синхронизации.

Лучший способ синхронизации — нажать «Синхронизировать всех пользователей» при первом создании поставщика LDAP, а затем настроить периодическую синхронизацию измененных пользователей.

LDAP-картографы

LDAP-картографы listenerszапускаются LDAP-провайдером. Они предоставляют еще одну точку расширения для интеграции LDAP. LDAP-картографы запускаются, когда:

- Пользователи входят в систему с помощью LDAP.

- Первоначально пользователи регистрируются.
- Консоль администратора запрашивает пользователя.

При создании поставщика LDAP Federation Tuxedo SSO автоматически предоставляет набор mappersдля этого поставщика. Этот набор может быть изменен пользователями, которые также могут разрабатывать мапперы или обновлять/удалять существующие.

Сопоставитель атрибутов пользователя

> Этот сопоставитель указывает, какой атрибут LDAP сопоставляется с атрибутом пользователя Tuxedo SSO. Например, вы можете настроить атрибут mailLDAP на emailатрибут в базе данных Tuxedo SSO. Для этой реализации сопоставителя всегда существует сопоставление один к одному.

Полное имя Mapper

> Этот сопоставитель указывает полное имя пользователя. Tuxedo SSO сохраняет имя в атрибуте LDAP (обычно cn) и сопоставляет имя с атрибутами firstNameи lastnameв базе данных Tuxedo SSO. Необходимость cnсодержать полное имя пользователя является обычной для развертываний LDAP.

Когда вы регистрируете новых пользователей в Tuxedo SSO и `Sync RegistrationsВКЛЮЧЕНО` для поставщика LDAP, средство сопоставления fullName позволяет вернуться к имени пользователя. Этот откат полезен при использовании Microsoft Active Directory (MSAD). Обычная настройка для MSAD заключается в настройке `cn`атрибута LDAP как fullName и одновременном использовании `cn`атрибута LDAP как `RDN LDAP Attribute`в конфигурации поставщика LDAP. При такой настройке Tuxedo SSO возвращается к имени пользователя. Например, если вы создаете пользователя Tuxedo SSO "john123" и оставляете firstName и lastName пустыми, то средство сопоставления fullname сохраняет "john123" как значение `cn`в LDAP. Когда вы позже вводите "John Doe" для firstName и lastName, средство сопоставления fullname обновляет LDAP `cn`до значения "John Doe", поскольку откат к имени пользователя не нужен.

Жестко закодированный картограф атрибутов

> Этот маппер добавляет жестко закодированное значение атрибута каждому пользователю Tuxedo SSO, связанному с LDAP. Этот маппер также может

принудительно устанавливать значения для свойств пользователя enabledили emailVerified.

Роль картографа

Этот сопоставитель настраивает сопоставления ролей из LDAP в сопоставления ролей Tuxedo SSO. Один сопоставитель ролей может сопоставлять роли LDAP (обычно группы из определенной ветви дерева LDAP) с ролями, соответствующими ролям области указанного клиента или ролям клиента. Вы можете настроить несколько сопоставителей ролей для того же поставщика LDAP. Например, вы можете указать, что сопоставления ролей из групп в ou=main,dc=example,dc=orgразделе сопоставления с сопоставлениями ролей области, а сопоставления ролей из групп в разделе ou=finance,dc=example,dc=orgсопоставления с сопоставлениями ролей клиента клиента finance.

Жестко закодированный картограф ролей

Этот сопоставитель предоставляет определенную роль Tuxedo SSO каждому пользователю Tuxedo SSO от поставщика LDAP.

Групповой картограф

Этот маппер сопоставляет группы LDAP из ветви дерева LDAP в группы внутри Tuxedo SSO. Этот маппер также распространяет сопоставления пользователей-групп из LDAP в сопоставления пользователей-групп в Tuxedo SSO.

MSAD User Account Mapper

Этот сопоставитель специфичен для Microsoft Active Directory (MSAD). Он может интегрировать состояние учетной записи пользователя MSAD в состояние учетной записи Tuxedo SSO, например, включенную учетную запись или просроченный пароль. Этот сопоставитель использует атрибуты userAccountControl, и pwdLastSetLDAP, специфичные для MSAD и не являющиеся стандартом LDAP. Например, если значение pwdLastSetравно 0, пользователь Tuxedo SSO должен обновить свой пароль. Результатом является обязательное действие UPDATE_PASSWORD, добавленное к пользователю. Если

значение userAccountControlравно 514(отключенная учетная запись), пользователь Tuxedo SSO отключен.

Сертификат картографа

Этот преобразователь отображает сертификаты X.509. Tuxedo SSO использует его в сочетании с аутентификацией X.509 и Full certificate in PEM formatв качестве источника удостоверений. Этот преобразователь ведет себя аналогично User Attribute Mapper, но Tuxedo SSO может фильтровать атрибут LDAP, хранящий сертификат формата PEM или DER. Включить Always Read Value From LDAPс этим преобразователем.

Сопоставители атрибутов пользователя, которые сопоставляют основные атрибуты пользователя Tuxedo SSO, такие как имя пользователя, имя, фамилия и адрес электронной почты, с соответствующими атрибутами LDAP. Вы можете расширить их и предоставить собственные дополнительные сопоставления атрибутов. Консоль администратора предоставляет подсказки, помогающие с настройкой соответствующих сопоставителей.

Хеширование паролей

Когда Tuxedo SSO обновляет пароль, Tuxedo SSO отправляет пароль в формате простого текста. Это действие отличается от обновления пароля во встроенной базе данных Tuxedo SSO, где Tuxedo SSO хэширует и добавляет соль к паролю перед отправкой в базу данных. Для LDAP Tuxedo SSO полагается на сервер LDAP для хэширования и добавления соли к паролю.

По умолчанию серверы LDAP, такие как MSAD, RHDS или FreeIPA, хэшируют и солят пароли. Другие серверы LDAP, такие как OpenLDAP или ApacheDS, хранят пароли в виде обычного текста, если только вы не используете расширенную операцию изменения пароля LDAPv3, как описано в RFC3062 . Включите расширенную операцию изменения пароля LDAPv3 на странице конфигурации LDAP. Более подробную информацию см. в документации вашего сервера LDAP.

Всегда проверяйте, что пароли пользователей правильно хэшированы и не хранятся в виде открытого текста, проверяя измененную запись каталога с помощью `ldapsearch`base64 и декодируя `userPassword`значение атрибута.

Настройка пула соединений

Для большей эффективности при управлении соединениями LDAP и для повышения производительности при обработке нескольких соединений можно включить пул соединений. Сделав это, когда соединение закрывается, оно будет возвращено в пул для будущего использования, тем самым снижая затраты на постоянное создание новых соединений.

Конфигурация пула соединений LDAP настраивается с использованием следующих системных свойств:

Имя	Описание
`com.sun.jndi.ldap.connect.pool.authentication`	Список типов аутентификации соединений, разделенных пробелами, которые могут быть объединены. Допустимые типы: "none", "simple" и "DIGEST-MD5"
`com.sun.jndi.ldap.connect.pool.initsize`	Строковое представление целого числа, которое представляет собой количество соединений на идентификатор соединения, создаваемых при первоначальном создании соединения для идентификатора.
`com.sun.jndi.ldap.connect.pool.maxsize`	Строковое представление целого числа, представляющего максимальное количество подключений на один идентификатор подключения, которые могут поддерживаться одновременно.
`com.sun.jndi.ldap.connect.pool.prefsize`	Строковое представление целого числа, представляющего предпочтительное количество подключений на один идентификатор подключения, которые должны

Имя	Описание
	поддерживаться одновременно.
com.sun.jndi.ldap.connect.pool.timeout	Строковое представление целого числа, представляющего собой количество миллисекунд, в течение которых неактивное соединение может оставаться в пуле, не будучи закрытым и удаленным из пула.
com.sun.jndi.ldap.connect.pool.protocol	Список разделенных пробелами типов протоколов соединений, которые могут быть объединены. Допустимые типы: "plain" и "ssl"
com.sun.jndi.ldap.connect.pool.debug	Строка, указывающая уровень отладочного вывода для создания. Допустимые значения: «fine» (создание и удаление трассировочного соединения) и «all» (вся отладочная информация)

Более подробную информацию см. в документации по настройке пула соединений Java LDAP .

Чтобы задать любое из этих свойств, вы можете задать JAVA_OPTS_APPENDпеременную среды:

export JAVA_OPTS_APPEND=-Dcom.sun.jndi.ldap.connect.pool.initsize=10 -Dcom.sun.jndi.ldap.connect.pool.maxsize=50

Поиск неисправностей

Полезно увеличить уровень ведения журнала до TRACE для категории org.Tuxedo SSO.storage.ldap. При такой настройке многие сообщения ведения журнала отправляются в журнал сервера на TRACEуровне, включая ведение журнала для всех запросов к серверу LDAP и параметры, которые использовались для отправки запросов. Когда вы создаете любой вопрос LDAP на форуме пользователя или в

JIRA, рассмотрите возможность присоединения журнала сервера с включенным ведением журнала TRACE. Если он слишком большой, хорошей альтернативой будет включить только фрагмент из журнала сервера с сообщениями, которые были добавлены в журнал во время операции, что вызывает у вас проблемы.

- При создании провайдера LDAP в журнале сервера на уровне INFO появляется сообщение, начинающееся с:

Creating new LDAP Store for the LDAP storage provider: ...

Он показывает конфигурацию вашего провайдера LDAP. Прежде чем задавать вопросы или сообщать об ошибках, будет неплохо включить это сообщение, чтобы показать вашу конфигурацию LDAP. В конце концов, не стесняйтесь заменять некоторые изменения конфигурации, которые вы не хотите включать, некоторыми значениями-заполнителями. Одним из примеров является bindDn=some-placeholder. Для connectionUrl, не стесняйтесь заменять его, но обычно полезно включать по крайней мере протокол, который использовался (ldapvs ldaps)`. Аналогично может быть полезно включить подробности для конфигурации ваших картографов LDAP, которые отображаются с таким сообщением на уровне DEBUG:

Mapper for provider: XXX, Mapper name: YYY, Provider: ZZZ ...

Обратите внимание, что эти сообщения отображаются только при включенном журнале DEBUG.

- Для отслеживания проблем производительности или пула подключений рассмотрите возможность установки значения свойства com.sun.jndi.ldap.connect.pool.debugна all. Это изменение добавляет множество дополнительных сообщений в журнал сервера с включенным журналированием для пула подключений LDAP. В результате вы можете отслеживать проблемы, связанные с пулом подключений или производительностью. Для получения более подробной информации см. Настройка пула подключений .

После изменения конфигурации пула соединений может потребоваться перезапустить сервер Tuxedo SSO, чтобы принудительно выполнить повторную инициализацию соединения с провайдером LDAP.

Если даже после перезапуска сервера сообщения о пуле соединений больше не появляются, это может означать, что пул соединений не работает с вашим сервером LDAP.

- В случае сообщения о проблеме LDAP вы можете рассмотреть возможность присоединения некоторой части вашего дерева LDAP с целевыми данными, которые вызывают проблемы в вашей среде. Например, если вход какого-либо пользователя занимает много времени, вы можете рассмотреть возможность присоединения его записи LDAP, показывающей количество memberатрибутов различных записей «группы». В этом случае может быть полезно добавить, сопоставлены ли эти записи групп с каким-либо средством сопоставления Group LDAP (или средством сопоставления Role LDAP) в Tuxedo SSO и т. д.

Интеграция управления идентификацией SSSD и FreeIPA

Tuxedo SSO включает в себя плагин System Security Services Daemon (SSSD) . SSSD является частью Fedora и Red Hat Enterprise Linux (RHEL) и обеспечивает доступ к нескольким поставщикам удостоверений и аутентификации. SSSD также предоставляет такие преимущества, как отказоустойчивость и поддержка в автономном режиме. Для получения дополнительной информации см. документацию по управлению удостоверениями Red Hat Enterprise Linux .

SSSD интегрируется с сервером управления идентификацией FreeIPA (IdM), обеспечивая аутентификацию и контроль доступа. Благодаря этой интеграции Tuxedo SSO может аутентифицироваться с помощью служб управления привилегированным доступом (PAM) и извлекать данные пользователей из SSSD. Для получения дополнительной информации об использовании Red Hat Identity Management в средах Linux см. документацию по Red Hat Enterprise Linux Identity Management .

Tuxedo SSO и SSSD взаимодействуют через интерфейсы D-Bus только для чтения. По этой причине для предоставления и обновления пользователей используется интерфейс администрирования FreeIPA/IdM. По умолчанию интерфейс импортирует имя пользователя, адрес электронной почты, имя и фамилию.

Tuxedo SSO автоматически регистрирует группы и роли, но не синхронизирует их. Любые изменения, внесенные администратором Tuxedo SSO в Tuxedo SSO, не синхронизируются с SSSD.

FreeIPA/IdM-сервер

Образ контейнера FreeIPA доступен на Quay.io. Чтобы настроить сервер FreeIPA, см. документацию FreeIPA .

Процедура

1. Запустите свой сервер FreeIPA с помощью этой команды:

   ```
   docker run --name freeipa-server-container -it \
   -h server.freeipa.local -e PASSWORD=YOUR_PASSWORD \
   -v /sys/fs/cgroup:/sys/fs/cgroup:ro \
   -v /var/lib/ipa-data:/data:Z freeipa/freeipa-server
   ```

   Параметр -hc server.freeipa.localпредставляет имя хоста сервера FreeIPA/IdM. Измените YOUR_PASSWORDпароль на свой собственный.

2. После запуска контейнера измените /etc/hostsфайл, включив в него:

   ```
   x.x.x.x     server.freeipa.local
   ```

   Если вы не внесете это изменение, вам придется настроить DNS-сервер.

3. Используйте следующую команду для регистрации вашего сервера Linux в домене IPA, чтобы поставщик федерации SSSD запускался и работал на Tuxedo SSO:

   ```
   ipa-client-install --mkhomedir -p admin -w password
   ```

4. Чтобы проверить работоспособность установки, выполните следующую команду на клиенте:

   ```
   kinit admin
   ```

5. Введите свой пароль.

6. Добавьте пользователей на сервер IPA с помощью этой команды:

$ ipa user-add <username> --first=<first name> --last=<surname> --email=<email address> --phone=<telephoneNumber> --street=<street> --city=<city> --state=<state> --postalcode=<postal code> --password

7. Принудительно установите пароль пользователя с помощью kinit.

   kinit <username>

8. Для восстановления нормальной работы IPA введите следующее:

   kdestroy -A
   kinit admin

SSSD и D-Bus

Поставщик федерации получает данные из SSSD с помощью D-BUS. Он аутентифицирует данные с помощью PAM.

Процедура

1. Установите RPM-пакет sssd-dbus.

   $ sudo yum install sssd-dbus

2. Запустите следующий скрипт подготовки:

   $ bin/federation-sssd-setup.sh

   Скрипт также можно использовать как руководство по настройке SSSD и PAM для Tuxedo SSO. Он вносит следующие изменения в /etc/sssd/sssd.conf:

   ```
   [domain/your-hostname.local]
   ...
   ldap_user_extra_attrs = mail:mail, sn:sn, givenname:givenname, telephoneNumber:telephoneNumber
   ...
   [sssd]
   services = nss, sudo, pam, ssh, ifp
   ...
   [ifp]
   allowed_uids = root, yourOSUsername
   user_attributes = +mail, +telephoneNumber, +givenname, +sn
   ```

Служба ifpдобавляется в SSSD и настраивается так, чтобы пользователь ОС мог опрашивать сервер IPA через этот интерфейс.

Скрипт также создает новую службу PAM /etc/pam.d/Tuxedo SSOдля аутентификации пользователей через SSSD:

```
auth    required   pam_sss.so
account required   pam_sss.so
```

3. Запустите, dbus-sendчтобы убедиться в успешности настройки.

```
dbus-send --print-reply --system --dest=org.freedesktop.sssd.infopipe
/org/freedesktop/sssd/infopipe org.freedesktop.sssd.infopipe.GetUserAttr
string:<username> array:string:mail,givenname,sn,telephoneNumber

dbus-send --print-reply --system --dest=org.freedesktop.sssd.infopipe
/org/freedesktop/sssd/infopipe org.freedesktop.sssd.infopipe.GetUserGroups
string:<username>
```

Если настройка прошла успешно, каждая команда отображает атрибуты и группы пользователя соответственно. Если есть тайм-аут или ошибка, поставщик федерации, работающий на Tuxedo SSO, не может получить никаких данных. Эта ошибка обычно происходит, потому что сервер не зарегистрирован на сервере FreeIPA IdM или не имеет разрешения на доступ к службе SSSD.

Если у вас нет разрешения на доступ к службе SSSD, убедитесь, что пользователь, запускающий сервер Tuxedo SSO, указан в /etc/sssd/sssd.confфайле в следующем разделе:

```
[ifp]
allowed_uids = root, yourOSUsername
```

И ipaapiсистемный пользователь создается внутри хоста. Этот пользователь необходим для ifpсервиса. Проверьте, что пользователь создан в системе.

```
grep ipaapi /etc/passwd
ipaapi:x:992:988:IPA Framework User:/:/sbin/nologin
```

Включение поставщика федерации SSSD

Tuxedo SSO использует проект DBus-Java для взаимодействия на низком уровне с D-Bus и JNA для аутентификации с помощью подключаемых модулей аутентификации операционной системы (PAM).

Хотя теперь Tuxedo SSO содержит все необходимые библиотеки для запуска SSSDпровайдера, требуется JDK версии 21. Поэтому SSSDпровайдер будет отображаться только в том случае, если конфигурация хоста корректна и для запуска Tuxedo SSO используется JDK 21.

Настройка федеративного хранилища SSSD

После установки настройте федеративное хранилище SSSD.

Процедура

1. Нажмите «Федерация пользователей» в меню.
2. Если все настроено успешно, на странице отобразится кнопка «Добавить поставщиков Sssd» . Нажмите на нее.
3. Присвойте имя новому поставщику.
4. Нажмите «Сохранить ».

Теперь вы можете пройти аутентификацию в Tuxedo SSO, используя пользователя и учетные данные FreeIPA/IdM.

Поставщики услуг на заказ

Tuxedo SSO имеет Service Provider Interface (SPI) для User Storage Federation для разработки пользовательских поставщиков. Вы можете найти документацию по разработке поставщиков клиентов в Server Developer Guide .

Управление пользователями

Из консоли администратора вы можете выполнять широкий спектр действий по управлению пользователями.

Создание пользователей

Вы создаете пользователей в области, где вы собираетесь иметь приложения, необходимые этим пользователям. Избегайте создания пользователей в главной области, которая предназначена только для создания других областей.

Предварительное условие

- Вы находитесь в мире, отличном от главного мира.

Процедура

1. Нажмите «Пользователи» в меню.
2. Нажмите Добавить пользователя .
3. Введите данные нового пользователя.

   Имя пользователя — единственное обязательное поле.
4. Нажмите Сохранить . После сохранения данных отобразится страница управления для нового пользователя.

Управление атрибутами пользователя

В Tuxedo SSO пользователь ассоциируется с набором атрибутов. Эти атрибуты используются для лучшего описания и идентификации пользователей в Tuxedo SSO, а также для передачи дополнительной информации о них приложениям.

Профиль пользователя определяет четко определенную схему для представления атрибутов пользователя и того, как они управляются в рамках области. Предоставляя согласованный вид информации о пользователе, он позволяет администраторам контролировать различные аспекты управления атрибутами, а также значительно упрощает расширение Tuxedo SSO для поддержки дополнительных атрибутов.

Хотя профиль пользователя в основном предназначен для атрибутов, которыми могут управлять конечные пользователи (например, имя и фамилия, телефон и т.

д.), он также служит для управления любыми другими метаданными, которые вы хотите связать со своими пользователями.

Помимо прочих возможностей, профиль пользователя позволяет администраторам:

- Определить схему для атрибутов пользователя
- Определите, требуется ли атрибут, на основе контекстной информации (например, требуется ли он только для пользователей или администраторов, или и для тех, и для других, или в зависимости от запрашиваемой области действия).
- Определите конкретные разрешения на просмотр и редактирование атрибутов пользователя, что позволит соблюдать строгие требования конфиденциальности, при которых некоторые атрибуты не могут быть просмотрены или изменены третьими лицами (включая администраторов).
- Динамически обеспечивать соответствие профиля пользователя требованиям, чтобы информация о пользователе всегда обновлялась и соответствовала метаданным и правилам, связанным с атрибутами.
- Определите правила проверки для каждого атрибута, используя встроенные валидаторы или написав собственные.
- Динамически отображайте формы, с которыми взаимодействуют пользователи, например, для регистрации, обновления профиля, брокерской деятельности и личной информации в консоли учетной записи, в соответствии с определениями атрибутов и без необходимости вручную менять темы.
- Настройте интерфейсы управления пользователями в консоли администрирования таким образом, чтобы атрибуты отображались динамически на основе схемы профиля пользователя.

Схема или конфигурация профиля пользователя использует формат JSON для представления атрибутов и их метаданных. Из консоли администрирования вы можете управлять конфигурацией, нажав на Realm Settingsменю слева, а затем нажав на User Profileвкладку на этой странице.

В следующих разделах мы рассмотрим, как создать собственную схему или конфигурацию профиля пользователя, а также как управлять атрибутами.

Понимание конфигурации по умолчанию

По умолчанию Tuxedo SSO предоставляет базовую конфигурацию профиля пользователя, охватывающую некоторые наиболее распространенные атрибуты пользователя:

Имя	Описание
`username`	Имя пользователя
`email`	Предпочтительный адрес электронной почты Конечного пользователя.
`firstName`	Имя(имена) или имя(имена) конечного пользователя
`lastName`	Фамилия(и) или имя(и) Конечного пользователя

В Tuxedo SSO оба атрибута usernameи emailимеют специальную обработку, поскольку они часто используются для идентификации, аутентификации и связывания учетных записей пользователей. Для этих атрибутов вы ограничены в изменении их настроек, и вы не можете их удалить.

> Поведение атрибутов `username`и `email`изменяется в соответствии с `Login`настройками вашей области. Например, изменение настроек `Email as username`или `Edit username`переопределит любую конфигурацию, которую вы установили в конфигурации профиля пользователя.

Как вы увидите в следующих разделах, вы можете свободно изменять конфигурацию по умолчанию, добавляя собственные атрибуты или изменяя настройки любого из доступных атрибутов, чтобы лучше соответствовать вашим потребностям.

Понимание контекстов профиля пользователя

В Tuxedo SSO управление пользователями осуществляется через различные контексты:

- Регистрация

- Обновить профиль
- Просмотр профиля при аутентификации через брокера или социального провайдера
- Консоль аккаунта
- Администрирование (например: консоль администрирования и API REST администратора)

За исключением Administrativeконтекста, все остальные контексты считаются контекстами конечного пользователя, поскольку они связаны с потоками самообслуживания пользователя.

Знание этих контекстов важно для понимания того, где конфигурация вашего профиля пользователя будет действовать при управлении пользователями. Независимо от контекста, в котором управляется пользователь, для отображения UI и проверки значений атрибутов будет использоваться одна и та же конфигурация профиля пользователя.

Как вы увидите в следующих разделах, вы можете ограничить определенные атрибуты, сделав их доступными только из административного контекста, и полностью отключить их для конечных пользователей. Обратное также верно, если вы не хотите, чтобы администраторы имели доступ к определенным атрибутам пользователя, а только конечный пользователь.

Понимание управляемых и неуправляемых атрибутов

По умолчанию Tuxedo SSO распознает только атрибуты, определенные в конфигурации вашего профиля пользователя. Сервер игнорирует любой другой атрибут, явно не определенный там.

Строго определяя, какие атрибуты пользователя могут быть установлены для ваших пользователей, а также как проверяются их значения, Tuxedo SSO может добавить еще один защитный барьер в вашу сферу и помочь вам предотвратить непредвиденные атрибуты и значения, связанные с вашими пользователями.

При этом атрибуты пользователя можно классифицировать следующим образом:

- Managed . Это атрибуты, контролируемые вашим профилем пользователя, которыми вы хотите разрешить конечным пользователям и администраторам управлять из любого контекста профиля пользователя. Для этих атрибутов вы хотите иметь полный контроль над тем, как и когда ими управляют.
- Неуправляемый . Это атрибуты, которые вы явно не определяете в своем профиле пользователя, поэтому они полностью игнорируются Tuxedo SSO по умолчанию.

Хотя неуправляемые атрибуты отключены по умолчанию, вы можете настроить свою область, используя различные политики, чтобы определить, как они обрабатываются сервером. Для этого щелкните в Realm Settingsменю слева, щелкните Generalвкладку, а затем выберите любой из следующих параметров в Unmanaged Attributesнастройках:

- Отключено . Это политика по умолчанию, при которой неуправляемые атрибуты отключены из всех контекстов профиля пользователя.
- Включено . Эта политика включает неуправляемые атрибуты для всех контекстов профилей пользователей.
- Администратор может просматривать . Эта политика включает неуправляемые атрибуты только из административного контекста как доступные только для чтения.
- Администратор может редактировать . Эта политика включает неуправляемые атрибуты только из административного контекста для чтения и записи.

Эти политики дают вам детальный контроль над тем, как сервер будет обрабатывать неуправляемые атрибуты. Вы можете полностью отключить или только поддерживать неуправляемые атрибуты при управлении пользователями через административный контекст.

Когда неуправляемые атрибуты включены (даже частично), вы можете управлять ими из консоли администрирования на Attributesвкладке в пользовательском интерфейсе User Details. Если политика установлена на Disabledэта вкладка недоступна.

В качестве рекомендации по безопасности старайтесь придерживаться максимально строгой политики (например: Disabledили Admin can edit), чтобы предотвратить установку неожиданных атрибутов (и значений) для ваших пользователей, когда они управляют своим профилем через контексты конечного пользователя. Избегайте установки Enabledполитики и предпочитайте определять все атрибуты, которыми конечные пользователи могут управлять в конфигурации вашего профиля пользователя, под вашим контролем.

> Политика `Enabled`предназначена для областей, мигрирующих с предыдущих версий Tuxedo SSO, а также для предотвращения нарушения поведения при использовании пользовательских тем и расширении сервера собственными пользовательскими атрибутами.

Как вы увидите в следующих разделах, вы также можете ограничить аудиторию для атрибута, выбрав, должен ли он быть видимым или доступным для записи пользователям и/или администраторам.

Для неуправляемых атрибутов максимальная длина составляет 2048 символов. Чтобы указать другую минимальную или максимальную длину, измените неуправляемый атрибут на управляемый атрибут и добавьте lengthвалидатор.

> Tuxedo SSO кэширует объекты, связанные с пользователем, во внутренних кэшах. Чем длиннее атрибуты, тем больше памяти потребляет кэш. Поэтому рекомендуется ограничить размер атрибутов длины. Рассмотрите возможность хранения больших объектов вне Tuxedo SSO и ссылайтесь на них по ID или URL.

Управление профилем пользователя

Конфигурация профиля пользователя управляется на основе области. Для этого щелкните ссылку Realm Settingsв меню слева, а затем щелкните User Profileвкладку.

Вкладка «Профиль пользователя»

На Attributesподвкладке представлен список всех управляемых атрибутов.

На Attribute Groupsподвкладке вы можете управлять группами атрибутов. Группа атрибутов позволяет вам сопоставлять атрибуты так, чтобы они отображались вместе при рендеринге форм, обращенных к пользователю.

На JSON Editorподвкладке вы можете просматривать и редактировать конфигурацию JSON . Вы можете использовать эту вкладку, чтобы получить

текущую конфигурацию или управлять ею вручную. Любые изменения, которые вы вносите на этой вкладке, отражаются на других вкладках, и наоборот.

В следующем разделе вы узнаете, как управлять атрибутами.

Управление атрибутами

На Attributesподвкладке вы можете создавать, редактировать и удалять управляемые атрибуты.

Чтобы определить новый атрибут и связать его с профилем пользователя, нажмите кнопку «Создать атрибут» в верхней части списка атрибутов.

Конфигурация атрибутов

При настройке атрибута вы можете определить следующие параметры:

Имя

> Имя атрибута, используемое для уникальной идентификации атрибута.

Отображаемое имя

> Удобное для пользователя имя атрибута, в основном используется при отображении форм, обращенных к пользователю. Также поддерживает использование интернационализированных сообщений

Многозначный

> Если включено, атрибут поддерживает несколько значений, и UI отображаются соответствующим образом, чтобы разрешить установку нескольких значений. При включении этого параметра обязательно добавьте валидатор, чтобы задать жесткое ограничение на количество значений.

Группа атрибутов

> Группа атрибутов, к которой принадлежит атрибут, если таковая имеется.

Включено, когда

> Включает или отключает атрибут. Если установлено значение Always, атрибут доступен из любого контекста профиля пользователя. Если установлено

значение Scopes are requested, атрибут доступен только тогда, когда клиент, действующий от имени пользователя, запрашивает набор из одной или нескольких областей. Эту опцию можно использовать для динамического применения определенных атрибутов в зависимости от запрашиваемых клиентских областей. Для консоли учетной записи и администрирования области не оцениваются, а атрибут всегда включен. Это связано с тем, что фильтрация атрибутов по областям работает только при запуске потоков аутентификации.

Необходимый

Установите условия, чтобы пометить атрибут как обязательный. Если отключено, атрибут необязателен. Если включено, вы можете установить настройку, Required forчтобы пометить атрибут как обязательный в зависимости от контекста профиля пользователя, чтобы атрибут был обязательным для конечных пользователей (через контексты конечных пользователей) или для администраторов (через административный контекст), или для обоих. Вы также можете установить настройку, Required whenчтобы пометить атрибут как обязательный только при запросе набора из одной или нескольких клиентских областей. Если установлено значение Always, атрибут требуется из любого контекста профиля пользователя. Если установлено значение Scopes are requested, атрибут требуется только тогда, когда клиент, действующий от имени пользователя, запрашивает набор из одной или нескольких областей. Для консолей учетной записи и администрирования области не оцениваются, и атрибут не требуется. Это связано с тем, что фильтрация атрибутов по областям работает только при запуске потоков аутентификации.

Разрешение

В этом разделе можно определить разрешения на чтение и запись, когда атрибут управляется из контекста конечного пользователя или административного контекста. Настройка Who can editпомечает атрибут как доступный для записи Userи/или Adminиз контекста конечного пользователя и административного контекста соответственно. Who can viewНастройка помечает атрибут как доступный только для чтения Userи/или Adminиз контекста конечного пользователя и административного контекста соответственно.

Проверка

> В этом разделе вы можете определить проверки, которые будут выполняться при управлении значением атрибута. Tuxedo SSO предоставляет набор встроенных валидаторов, из которых вы можете выбирать, с возможностью добавления собственных. Для получения более подробной информации см. раздел Проверка атрибутов .

Аннотация

> В этом разделе вы можете связать аннотации с атрибутом. Аннотации в основном полезны для передачи дополнительных метаданных на фронтенды для целей рендеринга. Для получения более подробной информации см. раздел Определение аннотаций пользовательского интерфейса .

Когда вы создаете атрибут, атрибут доступен только из административных контекстов, чтобы избежать неожиданного раскрытия атрибутов конечным пользователям. Фактически, атрибут не будет доступен конечным пользователям, когда они управляют своим профилем через контексты конечного пользователя. Вы можете изменить настройки Permissionsв любое время в соответствии с вашими потребностями.

Проверка атрибутов

Вы можете включить проверку управляемых атрибутов, чтобы убедиться, что значение атрибута соответствует определенным правилам. Для этого вы можете добавлять или удалять валидаторы из Validationsнастроек при управлении атрибутом.

Проверка атрибутов

Проверка выполняется в любой момент при записи в атрибут, и они могут выдавать ошибки, которые будут отображаться в пользовательских интерфейсах, если значение не проходит проверку.

По соображениям безопасности каждый атрибут, который может быть отредактирован пользователями, должен иметь проверку, чтобы ограничить размер вводимых пользователями значений. Если lengthпроверка не указана, Tuxedo SSO по умолчанию использует максимальную длину в 2048 символов.

Встроенные валидаторы

Tuxedo SSO предоставляет несколько встроенных валидаторов, из которых вы можете выбирать, а также вы можете предоставить свои собственные валидаторы, расширив Validator SPI.

Ниже представлен список всех встроенных валидаторов:

Имя	Описание	Конфигурация
длина	Проверьте длину строкового значения на основе минимальной и максимальной длины.	**min** : целое число, определяющее минимально допустимую длину. **max** : целое число, определяющее максимально допустимую длину. **trim-disabled** : логическое значение, определяющее, обрезается ли значение перед проверкой.
целое число	Проверьте, является ли значение целым числом и находится ли оно в пределах нижнего и/или верхнего диапазона. Если диапазон не определен, валидатор проверяет только, является ли значение допустимым числом.	**min** : целое число, определяющее нижний диапазон. **max** : целое число, определяющее верхний диапазон.
двойной	Проверьте, является ли значение числом double и находится ли оно в пределах нижнего и/или верхнего диапазона. Если диапазон не определен, валидатор проверяет только, является ли значение допустимым числом.	**min** : целое число, определяющее нижний диапазон. **max** : целое число, определяющее верхний диапазон.

Имя	Описание	Конфигурация
ури	Проверьте, является ли значение допустимым URI.	Никто
шаблон	Проверьте, соответствует ли значение определенному шаблону RegEx.	**шаблон** : шаблон RegEx, используемый при проверке значений. **error-message** : ключ сообщения об ошибке в i18n bundle. Если не задано, используется общее сообщение.
электронная почта	Проверьте, имеет ли значение допустимый формат электронной почты.	**max-local-length** : целое число для определения максимальной длины локальной части электронной почты. По умолчанию 64 согласно спецификации.
локальная дата	Проверьте, имеет ли значение допустимый формат с учетом области и/или локали пользователя.	Никто
изо-дата	Проверьте, имеет ли значение допустимый формат на основе ISO 8601. Этот валидатор можно использовать с входными данными, использующими тип ввода html5-date.	Никто
имя-человека-запрещенные-символы	Проверьте, является ли значение допустимым именем человека, как дополнительный барьер для атак, таких как	**error-message** : ключ сообщения об ошибке в i18n bundle. Если не задано, используется общее

Имя	Описание	Конфигурация
	внедрение скрипта. Проверка основана на шаблоне RegEx по умолчанию, который блокирует символы, нечасто встречающиеся в именах людей.	сообщение.
имя пользователя-запрещенные-символы	Проверьте, является ли значение допустимым именем пользователя, как дополнительный барьер для атак, таких как внедрение скрипта. Проверка основана на шаблоне RegEx по умолчанию, который блокирует символы, нечасто встречающиеся в именах пользователей. Когда `Email as username`включен параметр области, этот валидатор пропускается, чтобы разрешить значения электронной почты.	**error-message** : ключ сообщения об ошибке в i18n bundle. Если не задано, используется общее сообщение.
параметры	Проверьте, входит ли значение в определенный набор допустимых значений. Полезно для проверки значений, введенных через поля выбора и множественного выбора.	**параметры** : массив строк, содержащих допустимые значения.
вверх-имя-пользователя-не-idn-омограф	Поле может содержать только латинские символы и общие символы Unicode. Полезно для полей,	**error-message** : ключ сообщения об ошибке в i18n bundle. Если не задано, используется общее

Имя	Описание	Конфигурация
	которые могут быть объектом атак IDN-омографов (обычно имя пользователя).	сообщение.
многозначный	Проверяет размер многозначного атрибута.	**min** : целое число, определяющее минимально допустимое количество значений атрибута. **max** : целое число, определяющее максимально допустимое количество значений атрибутов.

Определение аннотаций пользовательского интерфейса

Чтобы передать дополнительную информацию на фронтенды, атрибуты могут быть украшены аннотациями, чтобы диктовать, как атрибуты будут отображаться. Эта возможность в основном полезна при расширении тем Tuxedo SSO для динамического отображения страниц на основе аннотаций, связанных с атрибутами.

Аннотации используются, например, для изменения HTML- typекода атрибута и изменения представления DOM атрибута , как вы увидите в следующих разделах.

Атрибут Аннотация

Аннотация — это пара ключ/значение, совместно используемая с пользовательским интерфейсом, чтобы они могли изменять способ отображения элемента HTML, соответствующего атрибуту. Вы можете задать любую аннотацию для атрибута, если она поддерживается темой, используемой в вашей области.

> Единственное ограничение, которое у вас есть, — это избегать использования аннотаций, использующих `kc`префикс в своих ключах, поскольку эти аннотации, использующие этот префикс, зарезервированы для Tuxedo SSO.

Встроенные аннотации

Встроенные темы Tuxedo SSO поддерживают следующие аннотации:

Имя	Описание
Тип ввода	Тип поля ввода формы. Доступные типы описаны в таблице ниже.
inputHelperTextBefore	Вспомогательный текст отображается перед (над) полем ввода. `${i18n.key}`Здесь можно использовать прямой текст или шаблон интернационализации (например). Текст НЕ экранируется html при отображении на странице, поэтому вы можете использовать html-теги здесь для форматирования текста, но вам также нужно правильно экранировать управляющие символы html.
inputHelperTextAfter	Вспомогательный текст отображается после (под) поля ввода. `${i18n.key}`Здесь можно использовать прямой текст или шаблон интернационализации (например). Текст НЕ экранируется html при отображении на странице, поэтому вы можете использовать html-теги здесь для форматирования текста, но вам также нужно правильно экранировать управляющие символы html.
inputOptionsFromValidation	Аннотация для типов select и multiselect. Необязательное имя проверки настраиваемого атрибута для получения входных параметров. Подробное описание см. ниже.
inputOptionLabelsI18nПрефикс	Аннотация для типов select и multiselect. Префикс ключа интернационализации для отображения параметров в пользовательском интерфейсе. Подробное описание см. ниже.
inputOptionМетки	Аннотация для типов select и multiselect. Необязательная карта для определения меток пользовательского интерфейса для опций (напрямую или с использованием интернационализации). Подробное описание см . ниже.
inputTypeЗаполнитель	HTML-атрибут input `placeholder`, применяемый к полю, — указывает короткую подсказку, описывающую ожидаемое значение поля ввода (например, пример значения или краткое описание ожидаемого формата). Краткая подсказка отображается в поле ввода до того, как пользователь введет

Имя	Описание
	значение.
inputTypeSize	Атрибут ввода HTML `size`, применяемый к полю, — указывает ширину в символах однострочного поля ввода. Для полей на основе `select`типа HTML указывает количество строк с отображаемыми параметрами. Может не работать, в зависимости от css в используемой теме!
inputTypeCols	HTML input `cols`атрибут, примененный к полю - указывает ширину в символах для `textarea`типа. Может не работать, в зависимости от css в используемой теме!
inputTypeRows	HTML-атрибут input `rows`, применяемый к полю, — указывает высоту в символах для `textarea`типа. Для выбранных полей указывает количество строк с отображаемыми параметрами. Может не работать, в зависимости от css в используемой теме!
inputTypePattern	Атрибут ввода HTML `pattern`, применяемый к полю, обеспечивающему проверку на стороне клиента — указывает регулярное выражение, по которому проверяется значение поля ввода. Полезно для однострочных вводов.
inputTypeMaxLength	Атрибут ввода HTML `maxlength`, применяемый к полю, обеспечивающему проверку на стороне клиента — максимальная длина текста, который можно ввести в поле ввода. Полезно для текстовых полей.
inputTypeMinLength	Атрибут ввода HTML `minlength`, применяемый к полю, обеспечивающему проверку на стороне клиента — минимальная длина текста, который можно ввести в поле ввода. Полезно для текстовых полей.
inputTypeMax	Атрибут ввода HTML `max`, применяемый к полю, обеспечивающему проверку на стороне клиента — максимальное значение, которое можно ввести в поле ввода. Полезно для числовых полей.
inputTypeMin	Атрибут ввода HTML `min`, применяемый к полю, обеспечивающему проверку на стороне клиента — минимальное значение, которое можно ввести в поле ввода.

Имя	Описание
	Полезно для числовых полей.
inputTypeStep	Атрибут ввода HTML `step`, примененный к полю — определяет интервал между допустимыми числами в поле ввода. Полезно для числовых полей.
Формат числа	Если установлено, `data-kcNumberFormat`атрибут добавляется к полю для форматирования значения на основе заданного формата. Эта аннотация предназначена для чисел, формат которых основан на количестве цифр, ожидаемых в определенной позиции. Например, формат `({2}) {5}-{4}`отформатирует значение поля как `(00) 00000-0000`.
Номер неформатирован	Если установлено, `data-kcNumberUnFormat`атрибут добавляется в поле для форматирования значения на основе заданного формата перед отправкой формы. Эта аннотация полезна, если вы не хотите сохранять какой-либо формат для определенного атрибута, а только форматируете значение на стороне клиента. Например, если текущее значение равно `(00) 00000-0000`, значение изменится на , `00000000000`если вы установите значение `{11}`для этой аннотации или любого другого нужного вам формата, указав набор из одной или нескольких групп цифр. Обязательно добавьте валидаторы для выполнения валидации на стороне сервера перед сохранением значений.

Типы полей используют теги полей HTML-формы и атрибуты, применяемые к ним. Они ведут себя на основе спецификаций HTML и поддержки их браузерами.

Визуальная отрисовка также зависит от стилей CSS, примененных в используемой теме.

Изменение HTML typeдля атрибута

Вы можете изменить typeэлемент ввода HTML5, установив inputTypeаннотацию. Доступны следующие типы:

Имя	Описание	HTML-тег используется
текст	Ввод текста в одну строку.	вход
текстовая область	Ввод многострочного текста.	текстовая область

Имя	Описание	HTML-тег используется
выбирать	Общий одиночный выбор входа. Смотрите описание, как настроить параметры ниже.	выбирать
выберите-радиокнопки	Одиночный выбор входа через группу радиокнопок. Смотрите описание, как настроить параметры ниже.	группа ввода
множественный выбор	Общий множественный выбор ввода. Смотрите описание настройки опций ниже.	выбирать
множественный выбор-флажков	Множественный выбор ввода через группу флажков. Смотрите описание, как настроить параметры ниже.	группа ввода
html5-электронная почта	Однострочный ввод текста для адреса электронной почты на основе спецификации HTML 5.	вход
html5-тел	Однострочный ввод текста для номера телефона на основе спецификации HTML 5.	вход
html5-url	Ввод текста в одну строку для URL на основе спецификации HTML 5.	вход
html5-номер	Ввод числа в одну строку (целое или с плавающей точкой в зависимости от `step`) на основе спецификации HTML 5.	вход
html5-диапазон	Ползунок для ввода чисел на основе спецификации HTML 5.	вход
html5-дата-время-локальный	Ввод даты и времени на основе спецификации HTML 5.	вход
html5-дата	Ввод даты на основе спецификации HTML 5.	вход

Имя	Описание	HTML-тег используется
html5-месяцев	Ввод месяца на основе спецификации HTML 5.	вход
html5-недельный	Недельный ввод на основе спецификации HTML 5.	вход
html5-время	Ввод времени на основе спецификации HTML 5.	вход

Определение параметров для полей выбора и множественного выбора
Параметры для полей select и multiselect берутся из проверки, применяемой к атрибуту, чтобы гарантировать, что параметры проверки и поля, представленные в пользовательском интерфейсе, всегда согласованы. По умолчанию параметры берутся из встроенной optionsпроверки.

Вы можете использовать различные способы, чтобы предоставить удобные для восприятия человеком метки для опций select и multiselect. Самый простой случай — когда значения атрибутов совпадают с метками пользовательского интерфейса. В этом случае дополнительная настройка не требуется.

Значения параметров совпадают с метками пользовательского интерфейса.

Когда значение атрибута — это идентификатор, не подходящий для пользовательского интерфейса, можно использовать простую поддержку интернационализации, предоставляемую inputOptionLabelsI18nPrefixаннотацией. Она определяет префикс для ключей интернационализации, значение параметра — точка, добавленная к этому префиксу.

Простая интернационализация меток пользовательского интерфейса с использованием ключевого префикса i18n

userprofile.jobtitle.swengЗатем с помощью ключей и необходимо предоставить локализованные тексты меток пользовательского интерфейса для значений параметров userprofile.jobtitle.swarch, используя общий механизм локализации.

Вы также можете использовать inputOptionLabelsаннотацию для предоставления меток для отдельных опций. Она содержит карту меток для опции - ключ в карте - это значение опции (определенное при проверке), а значение в карте - это сам

текст метки пользовательского интерфейса или его шаблон интернационализации (например, ${i18n.key}) для этой опции.

`JSON Editor`Для ввода карты в качестве значения аннотации необходимо использовать профиль пользователя `inputOptionLabels`.

Пример напрямую введенных меток для отдельных опций без интернационализации:

```
"attributes": [
<...
{
  "name": "jobTitle",
  "validations": {
    "options": {
      "options":[
        "sweng",
        "swarch"
      ]
    }
  },
  "annotations": {
    "inputType": "select",
    "inputOptionLabels": {
      "sweng": "Software Engineer",
      "swarch": "Software Architect"
    }
  }
}
...
]
```

Пример интернационализированных меток для отдельных опций:

```
"attributes": [
...
{
  "name": "jobTitle",
  "validations": {
```

```
    "options": {
      "options":[
        "sweng",
        "swarch"
      ]
    }
  },
  "annotations": {
    "inputType": "select-radiobuttons",
    "inputOptionLabels": {
      "sweng": "${jobtitle.swengineer}",
      "swarch": "${jobtitle.swarchitect}"
    }
  }
}
...
]
```

jobtitle.swengineerЗатем локализованные тексты и ключи должны быть предоставлены jobtitle.swarchitectc использованием общего механизма локализации.

Пользовательский валидатор может использоваться для предоставления опций благодаря inputOptionsFromValidationаннотации атрибутов. Эта валидация должна иметь optionsконфигурацию, предоставляющую массив опций. Интернационализация работает так же, как и для опций, предоставляемых встроенной optionsвалидацией.

Параметры, предоставляемые пользовательским валидатором

Изменение представления DOM атрибута
Вы можете включить дополнительное поведение на стороне клиента, установив аннотации с kcпрефиксом. Эти аннотации будут преобразованы в атрибут HTML в соответствующем элементе атрибута, с префиксом data-, и скрипт с тем же именем будет загружен на динамические страницы, чтобы вы могли выбирать элементы из DOM на основе пользовательского data-атрибута и декорировать их соответствующим образом, изменяя их представление DOM.

Например, если вы добавляете kcMyCustomValidationаннотацию к атрибуту, атрибут HTML data-kcMyCustomValidationдобавляется к соответствующему элементу HTML для атрибута, а модуль JavaScript загружается из вашей пользовательской темы в <THEME TYPE>/resources/js/kcMyCustomValidation.js. См. Руководство разработчика сервера для получения дополнительной информации о том, как развернуть пользовательский модуль JavaScript в вашей теме.

Модуль JavaScript может запустить любой код для настройки DOM и элементов, отображаемых для каждого атрибута. Для этого вы можете использовать userProfile.jsмодуль для регистрации дескриптора аннотации для вашей пользовательской аннотации следующим образом:

```
import { registerElementAnnotatedBy } from "./userProfile.js";

registerElementAnnotatedBy({
  name: 'kcMyCustomValidation',
  onAdd(element) {
    var listener = function (event) {
        // do something on keyup
    };

    element.addEventListener("keyup", listener);

    // returns a cleanup function to remove the event listener
    return () => element.removeEventListener("keyup", listener);
  }
});
```

Это registerElementAnnotatedByуметод регистрации дескрипторов аннотаций. Дескриптор — это объект с name, ссылающийся на имя аннотации, и onAddфункцией. Всякий раз, когда страница отображается или атрибут с аннотацией добавляется в DOM, onAddвызывается функция, чтобы вы могли настроить поведение элемента.

Функция onAddтакже может возвращать функцию для выполнения очистки. Например, если вы добавляете прослушиватели событий к элементам, вы можете захотеть удалить их в случае, если элемент будет удален из DOM.

Кроме того, вы также можете использовать любой код JavaScript, если его userProfile.jsнедостаточно для ваших нужд:

```
document.querySelectorAll(`[data-kcMyCustomValidation]`).forEach((element) => {
    var listener = function (evt) {
        // do something on keyup
    };

    element.addEventListener("keyup", listener);
});
```

Управление группами атрибутов

На Attribute Groupsподвкладке вы можете создавать, редактировать и удалять группы атрибутов. Группа атрибутов позволяет вам определить контейнер для коррелированных атрибутов, чтобы они отображались вместе в формах, обращенных к пользователю.

Список групп атрибутов

> Вы не можете удалить группы атрибутов, которые привязаны к атрибутам. Для этого вам следует сначала обновить атрибуты, чтобы удалить привязку.

Чтобы создать новую группу, нажмите кнопку «Создать группу атрибутов» в верхней части списка групп атрибутов.

Конфигурация группы атрибутов

При настройке группы вы можете определить следующие параметры:

Имя

> Имя атрибута, используемое для уникальной идентификации атрибута.

Отображаемое имя

> Удобное для пользователя имя атрибута, в основном используется при отображении форм, обращенных к пользователю. Также поддерживает использование интернационализированных сообщений

Показать описание

Удобный для пользователя текст, который будет отображаться как подсказка при отображении форм, обращенных к пользователю. Он также поддерживает использование интернационализированных сообщений

Аннотация

В этом разделе вы можете связать аннотации с атрибутом. Аннотации в основном полезны для передачи дополнительных метаданных во фронтенды для целей рендеринга.

Использование конфигурации JSON

Конфигурация профиля пользователя хранится с использованием четко определенной схемы JSON. Вы можете выбрать редактирование конфигурации профиля пользователя напрямую, нажав на JSON Editorвложенную вкладку.

Конфигурация JSON

Схема JSON определяется следующим образом:

```
{
  "unmanagedAttributePolicy": "DISABLED",
  "attributes": [
    {
      "name": "myattribute",
      "multivalued": false,
      "displayName": "My Attribute",
      "group": "personalInfo",
      "required": {
        "roles": [ "user", "admin" ],
        "scopes": [ "foo", "bar" ]
      },
      "permissions": {
        "view": [ "admin", "user" ],
        "edit": [ "admin", "user" ]
      },
      "validations": {
        "email": {
          "max-local-length": 64
        },
```

```
        "length": {
          "max": 255
        }
      },
      "annotations": {
        "myannotation": "myannotation-value"
      }
    }
  ],
  "groups": [
    {
      "name": "personalInfo",
      "displayHeader": "Personal Information",
      "annotations": {
        "foo": ["foo-value"],
        "bar": ["bar-value"]
      }
    }
  ]
}
```

Схема поддерживает столько атрибутов и групп, сколько вам необходимо.

Свойство unmanagedAttributePolicyопределяет политику неуправляемых атрибутов, устанавливая одно из следующих значений. Для получения более подробной информации см. Understanding Managed and Unmanaged Attributes .

- DISABLED
- ENABLED
- ADMIN_VIEW
- ADMIN_EDIT

Схема атрибутов
Для каждого атрибута необходимо определить nameи, по желанию, requiredпараметры permission, и annotations.

Свойство requiredопределяет, является ли атрибут обязательным. Tuxedo SSO позволяет вам устанавливать атрибут как требуемый на основе различных условий.

Если requiredсвойство определено как пустой объект, атрибут всегда обязателен.

```
{
  "attributes": [
    {
      "name": "myattribute",
      "required": {}
  ]
}
```

С другой стороны, вы можете сделать атрибут обязательным только для пользователей, администраторов или и тех, и других. А также пометить атрибут как обязательный только в случае, если запрашивается определенная область действия при аутентификации пользователя в Tuxedo SSO.

Чтобы отметить атрибут как обязательный для пользователя и/или администратора, задайте rolesсвойство следующим образом:

```
{
  "attributes": [
    {
      "name": "myattribute",
      "required": {
        "roles": ["user"]
      }
  ]
}
```

Свойство rolesожидает массив, значениями которого могут быть userили admin, в зависимости от того, требуется ли атрибут пользователю или администратору соответственно.

Аналогично, вы можете сделать атрибут обязательным, когда набор из одной или нескольких областей запрашивается клиентом при аутентификации пользователя. Для этого вы можете использовать свойство scopesследующим образом:

```
{
  "attributes": [
    {
      "name": "myattribute",
      "required": {
        "scopes": ["foo"]
      }
  ]
}
```

Свойство scopesпредставляет собой массив, значениями которого могут быть любые строки, представляющие область действия клиента.

Свойство уровня атрибута permissionsможет использоваться для определения разрешений на чтение и запись атрибута. Разрешения устанавливаются на основе того, могут ли эти операции выполняться над атрибутом пользователем, администратором или обоими.

```
{
  "attributes": [
    {
      "name": "myattribute",
      "permissions": {
        "view": ["admin"],
        "edit": ["user"]
      }
  ]
}
```

Оба свойства viewи editожидают массив, значениями которого могут быть userили admin, в зависимости от того, доступен ли атрибут для просмотра или редактирования пользователю или администратору соответственно.

Когда editразрешение предоставляется, оно viewпредоставляется неявно.

Свойство уровня атрибута annotationможет использоваться для связывания дополнительных метаданных с атрибутами. Аннотации в основном полезны для передачи дополнительной информации об атрибутах во внешние интерфейсы,

отображающие атрибуты пользователя на основе конфигурации профиля пользователя. Каждая аннотация представляет собой пару ключ/значение.

```
{
  "attributes": [
    {
      "name": "myattribute",
      "annotations": {
        "foo": ["foo-value"],
        "bar": ["bar-value"]
      }
  ]
}
```

Схема группы атрибутов
Для каждой группы атрибутов необходимо определить nameи, при необходимости, annotationsнастройки.

Свойство уровня атрибута annotationможет использоваться для связывания дополнительных метаданных с атрибутами. Аннотации в основном полезны для передачи дополнительной информации об атрибутах во внешние интерфейсы, отображающие атрибуты пользователя на основе конфигурации профиля пользователя. Каждая аннотация представляет собой пару ключ/значение.

Настройка отображения пользовательских интерфейсов

Пользовательские интерфейсы всех контекстов профилей пользователей (включая консоль администрирования) отображаются динамически в соответствии с конфигурацией вашего профиля пользователя.

Механизм рендеринга по умолчанию предоставляет следующие возможности:

- Показывать или скрывать поля в зависимости от разрешений, установленных для атрибутов.
- Отображать маркеры для обязательных полей на основе ограничений, установленных для атрибутов.

- Измените тип ввода поля (текст, дата, число, выбор, множественный выбор), задав атрибут.
- Пометьте поля как доступные только для чтения в зависимости от разрешений, установленных для атрибута.
- Упорядочить поля в зависимости от порядка, заданного для атрибутов.
- Группируйте поля, принадлежащие к одной и той же группе атрибутов.
- Динамически группируйте поля, принадлежащие к одной группе атрибутов.

Атрибуты упорядочивания
Порядок атрибутов задается путем перетаскивания строк атрибутов на странице списка атрибутов.

Атрибуты упорядочения

Порядок, установленный вами на этой странице, соблюдается при отображении полей в динамических формах.

Группировка атрибутов
При отображении динамических форм они будут пытаться сгруппировать атрибуты, принадлежащие к одной и той же группе атрибутов.

Форма динамического обновления профиля

Когда атрибуты связаны с группой атрибутов, порядок атрибутов также важен, чтобы атрибуты в одной группе располагались близко друг к другу, в одном заголовке группы. В противном случае, если атрибуты в группе не имеют последовательного порядка, вы можете получить один и тот же заголовок группы, отображаемый несколько раз в динамической форме.

Включение прогрессивного профилирования

Чтобы убедиться, что профили конечных пользователей соответствуют конфигурации, администраторы могут использовать VerifyProfiletребуемое действие, чтобы в конечном итоге заставить пользователей обновить свои профили при аутентификации в Tuxedo SSO.

Действие `VerifyProfile`похоже на `UpdateProfile`действие. Однако оно использует все возможности, предоставляемые профилем пользователя, для автоматического обеспечения

соответствия конфигурации профиля пользователя.

Если эта функция включена, VerifyProfileдействие будет выполнять следующие шаги при аутентификации пользователя:

- Проверьте, полностью ли профиль пользователя соответствует конфигурации профиля пользователя, установленной для области. Это означает запуск валидаций и убедитесь, что все они успешны.
- Если нет, выполните дополнительный шаг во время аутентификации, чтобы пользователь мог обновить любой отсутствующий или недействительный атрибут.
- Если профиль пользователя соответствует конфигурации, никаких дополнительных шагов не выполняется, и пользователь продолжает процесс аутентификации.

Действие VerifyProfilevключено по умолчанию. Чтобы отключить его, щелкните Authenticationссылку в меню слева, а затем щелкните Required Actionsвкладку. На этой вкладке используйте
переключатель Включено действия, VerifyProfileчтобы отключить его.

Регистрация VerifyProfile Требуемое действие

Использование интернационализированных сообщений

Если вы хотите использовать интернационализированные сообщения при настройке атрибутов, групп атрибутов и аннотаций, вы можете задать их отображаемое имя, описание и значения, используя заполнитель, который будет преобразован в сообщение из пакета сообщений.

Для этого вы можете использовать заполнитель для разрешения ключей сообщений, например ${myAttributeName}, где myAttributeNameнаходится ключ для сообщения в пакете сообщений. Для получения более подробной информации см. Руководство разработчика сервера о том, как добавлять пакеты сообщений в пользовательские темы.

Определение учетных данных пользователя

Управлять учетными данными пользователя можно на вкладке «Учетные данные» .

Управление учетными данными

Вы меняете приоритет учетных данных, перетаскивая строки. Новый порядок определяет приоритет учетных данных для этого пользователя. Верхний учетный параметр имеет наивысший приоритет. Приоритет определяет, какой учетный параметр отображается первым после входа пользователя в систему.

Тип

> В этом столбце отображается тип учетных данных, например пароль или одноразовый пароль .

Метка пользователя

> Это назначаемая метка для распознавания учетных данных при представлении в качестве опции выбора во время входа в систему. Она может быть установлена на любое значение для описания учетных данных.

Данные

> Это неконфиденциальная техническая информация об учетных данных. По умолчанию она скрыта. Вы можете нажать Показать данные…, чтобы отобразить данные для учетных данных.

Действия

> Нажмите «Сбросить пароль» , чтобы изменить пароль пользователя, и «Удалить» , чтобы удалить учетные данные.

Вы не можете настроить другие типы учетных данных для конкретного пользователя в консоли администратора; эта задача является обязанностью пользователя.

Вы можете удалить учетные данные пользователя в случае, если пользователь потеряет устройство OTP или если учетные данные были скомпрометированы. Удалить учетные данные пользователя можно только на вкладке Credentials .

Установка пароля для пользователя

Если у пользователя нет пароля или пароль был удален, отображается раздел «Установить пароль» .

Если у пользователя уже есть пароль, его можно сбросить в разделе «Сброс пароля» .

Процедура

1. Нажмите Пользователи в меню. Отобразится страница Пользователи .
2. Выберите пользователя.
3. Перейдите на вкладку «Учетные данные» .
4. Введите новый пароль в разделе «Установить пароль» .
5. Нажмите «Установить пароль» .

   Если **Temporary** имеет **значение ON** , пользователь должен сменить пароль при первом входе в систему. Чтобы разрешить пользователям сохранять предоставленный пароль, установите **Temporary** в **значение OFF.** Пользователь должен нажать **Set Password,** чтобы сменить пароль.

Запрос на сброс пароля пользователем

Вы также можете попросить пользователя сбросить пароль.

Процедура

1. Нажмите Пользователи в меню. Отобразится страница Пользователи .
2. Выберите пользователя.
3. Перейдите на вкладку «Учетные данные» .
4. Нажмите «Сброс учетных данных» .
5. Выберите Обновить пароль из списка.
6. Нажмите Отправить письмо . Отправленное письмо содержит ссылку, которая направляет пользователя в окно обновления пароля .

7. При желании вы можете установить срок действия ссылки электронной почты. Это установлено по умолчанию на вкладке Токены в Настройках Realm .

Создание одноразового пароля

Если OTP является условным в вашей области, пользователь должен перейти в Tuxedo SSO Account Console, чтобы перенастроить новый генератор OTP. Если OTP является обязательным, пользователь должен перенастроить новый генератор OTP при входе в систему.

В качестве альтернативы вы можете отправить пользователю электронное письмо с просьбой сбросить генератор OTP. Следующая процедура также применима, если у пользователя уже есть учетные данные OTP.

Предварительное условие

- Вы вошли в соответствующую область.

Процедура

1. Нажмите Пользователи в главном меню. Отобразится страница Пользователи .
2. Выберите пользователя.
3. Перейдите на вкладку «Учетные данные» .
4. Нажмите «Сброс учетных данных» .
5. Установите действия по сбросу для настройки OTP .
6. Нажмите Отправить письмо . Отправленное письмо содержит ссылку, которая направляет пользователя на страницу настройки OTP .

Разрешение пользователям самостоятельно регистрироваться

Вы можете использовать Tuxedo SSO как сторонний сервер авторизации для управления пользователями приложения, включая пользователей, которые

регистрируются самостоятельно. Если вы включите самостоятельную регистрацию, на странице входа отобразится ссылка на регистрацию, чтобы пользователь мог создать учетную запись.

Ссылка для регистрации

Пользователь должен добавить информацию профиля в регистрационную форму для завершения регистрации. Регистрационную форму можно настроить, удалив или добавив поля, которые должен заполнить пользователь.

Разъяснение по брокерской идентификации и административному API

Даже если самостоятельная регистрация отключена, новых пользователей можно добавлять в Tuxedo SSO одним из следующих способов:

- Администратор может добавлять новых пользователей с помощью консоли администратора (или REST API администратора)
- Когда включено посредничество идентификации, новые пользователи, аутентифицированные поставщиком идентификации, могут автоматически добавляться/регистрироваться в хранилище Tuxedo SSO. Для получения дополнительной информации см. раздел «Первый процесс входа» в главе «Посредничество идентификации» .

Кроме того, пользователи, поступающие из стороннего хранилища пользователей (например, LDAP), автоматически доступны в Tuxedo SSO, когда включено соответствующее хранилище пользователей.

Дополнительные ресурсы

- Дополнительную информацию о настройке регистрации пользователей см. в Руководстве разработчика сервера .

Включение регистрации пользователя

Предоставьте пользователям возможность самостоятельной регистрации.

Процедура

1. Нажмите «Настройки области» в главном меню.

2. Нажмите вкладку «Вход» .
3. Включите регистрацию пользователей .

После включения этого параметра на странице входа в консоль администратора отобразится ссылка «Регистрация» .

Регистрация нового пользователя

Как новый пользователь, вы должны заполнить регистрационную форму, чтобы войти в систему в первый раз. Вы добавляете информацию профиля и пароль для регистрации.

Регистрационная форма

Предварительное условие

- Регистрация пользователей включена.

Процедура

1. Нажмите ссылку «Регистрация» на странице входа. Отобразится страница регистрации.
2. Введите информацию профиля пользователя.
3. Введите новый пароль.
4. Нажмите «Зарегистрироваться» .

Требование от пользователя согласиться с условиями во время регистрации

Для регистрации пользователя вы можете потребовать согласие с вашими условиями.

Регистрационная форма с обязательным соглашением о положениях и условиях

Предварительное условие

- Регистрация пользователей включена.

- Включено действие, требующее соблюдения положений и условий.

Процедура

1. Нажмите «Аутентификация» в меню. Нажмите вкладку «Потоки» .
2. Нажмите на ссылку «Регистрация» .
3. Выберите «Обязательно» в строке «Условия» .

   Согласуйте условия и положения, требуемые при регистрации.

Определение действий, необходимых при входе в систему

Вы можете задать действия, которые пользователь должен выполнить при первом входе в систему. Эти действия требуются после того, как пользователь предоставит учетные данные. После первого входа в систему эти действия больше не требуются. Вы добавляете требуемые действия на вкладке Details этого пользователя.

Некоторые требуемые действия автоматически запускаются для пользователя во время входа в систему, даже если они явно не добавлены администратором к этому пользователю. Например, Update passwordдействие может быть запущено, если политики паролей настроены таким образом, что пароль пользователя необходимо менять каждые X дней. Или verify profileдействие может потребовать от пользователя обновить профиль пользователя , пока некоторые атрибуты пользователя не будут соответствовать требованиям согласно конфигурации профиля пользователя.

Ниже приведены примеры требуемых типов действий:

Обновить пароль

Пользователь должен сменить свой пароль.

Настроить одноразовый пароль

Пользователь должен настроить генератор одноразовых паролей на своем мобильном устройстве с помощью приложения Free OTP или Google Authenticator.

Подтвердить адрес электронной почты

Пользователь должен подтвердить свой адрес электронной почты. Пользователю будет отправлено электронное письмо со ссылкой для проверки, по которой он должен щелкнуть. После успешного завершения этого рабочего процесса пользователю будет разрешено войти в систему.

Обновить профиль

Пользователь должен обновить информацию профиля, такую как имя, адрес, адрес электронной почты и номер телефона.

Некоторые действия не имеет смысла добавлять в учетную запись пользователя напрямую. Например, это `Update User Locale`вспомогательное действие для обработки некоторых параметров, связанных с локализацией. Другим примером является действие `Delete Credential`, которое должно запускаться как параметризованный AIA . Что касается этого, если администратор хочет удалить учетные данные какого-либо пользователя, он может сделать это напрямую в консоли администратора. Действие `Delete Credential`предназначено для использования, например, консолью учетной записи Tuxedo SSO .

Настройка требуемых действий для одного пользователя

Вы можете задать действия, которые требуются для любого пользователя.

Процедура

1. Нажмите «Пользователи» в меню.
2. Выберите пользователя из списка.
3. Перейдите к списку «Требуемые действия пользователя» .
4. Выберите все действия, которые вы хотите добавить в учетную запись.
5. Нажмите X рядом с названием действия, чтобы удалить его.
6. Нажмите «Сохранить» после того, как выберете действия для добавления.

Настройка обязательных действий для всех пользователей

Вы можете указать, какие действия требуются перед первым входом всех новых пользователей. Требования применяются к пользователю, созданному кнопкой Добавить пользователя на странице Пользователи или ссылкой Регистрация на странице входа.

Процедура

1. Нажмите «Аутентификация» в меню.
2. Перейдите на вкладку «Требуемые действия» .
3. Установите флажок в столбце Установить как действие по умолчанию для одного или нескольких требуемых действий. Когда новый пользователь входит в систему в первый раз, выбранные действия должны быть выполнены.

Включение положений и условий в качестве обязательного действия

Вы можете включить обязательное действие, согласно которому новые пользователи должны принять положения и условия перед первым входом в Tuxedo SSO.

Процедура

1. Нажмите «Аутентификация» в меню.
2. Перейдите на вкладку «Требуемые действия» .
3. Включите действие «Условия и положения» .
4. Отредактируйте terms.ftlфайл в базовой теме входа.

Дополнительные ресурсы

- Дополнительную информацию о расширении и создании тем см. в Руководстве разработчика сервера .

Действия, инициированные приложением

Действия, инициированные приложением (AIA), позволяют клиентским приложениям запрашивать у пользователя выполнение действия на стороне Tuxedo SSO. Обычно, когда клиентское приложение OIDC хочет, чтобы пользователь вошел в систему, оно перенаправляет этого пользователя на URL-адрес входа, как описано в разделе OIDC . После входа пользователь перенаправляется обратно в клиентское приложение. Пользователь выполняет действия, которые требовал администратор, как описано в предыдущем разделе , а затем немедленно перенаправляется обратно в приложение. Однако AIA позволяет клиентскому приложению запрашивать некоторые требуемые действия у пользователя во время входа в систему. Это можно сделать, даже если пользователь уже аутентифицирован на клиенте и имеет активный сеанс SSO. Он запускается путем добавления kc_actionпараметра к URL-адресу входа OIDC со значением, содержащим запрошенное действие.
Например, kc_action=UPDATE_PASSWORDпараметр.

Пользователь может отменить действие, инициированное приложением. В этом случае пользователь перенаправляется обратно в клиентское приложение. URI перенаправления будет содержать параметры
запроса kc_action_status=cancelledи kc_actionимя отмененного действия.

> Параметры `kc_action`и `kc_action_status`представляют собой фирменный механизм Tuxedo SSO, не поддерживаемый спецификацией OIDC.
>
> Действия, инициированные приложением, поддерживаются только для клиентов OIDC.

Таким образом, если используется AIA, пример потока будет аналогичен следующему:

- Клиентское приложение перенаправляет пользователя на URL-адрес входа в OIDC с дополнительным параметром,
  например:kc_action=UPDATE_PASSWORD
- Всегда запускается browserпоток, как описано в разделе Потоки аутентификации . Если пользователь не был аутентифицирован, ему необходимо пройти аутентификацию, как при обычном входе в систему. Если пользователь уже был аутентифицирован, этот пользователь может

быть автоматически повторно аутентифицирован с помощью файла cookie SSO без необходимости активной повторной аутентификации и повторного предоставления учетных данных. В этом случае пользователь будет напрямую перенаправлен на экран с определенным действием (в данном случае — обновить пароль). Однако в некоторых случаях требуется активная повторная аутентификация, даже если у пользователя есть файл cookie SSO (подробности см. ниже).

- Экран с определенным действием (в данном случае update password) отображается пользователю, чтобы он мог выполнить определенное действие.
- Затем пользователь перенаправляется обратно в клиентское приложение.

Обратите внимание, что AIA используются консолью учетных записей Tuxedo SSO для запроса обновления пароля или сброса других учетных данных, таких как OTP или WebAuthn.

> Даже если параметр `kc_action`был использован, недостаточно предполагать, что пользователь всегда выполняет действие. Например, пользователь мог вручную удалить параметр `kc_action`из URL-адреса браузера. Поэтому нет гарантии, что у пользователя есть OTP для учетной записи после того, как клиент запросил `kc_action=CONFIGURE_TOTP`. Если вы хотите проверить, что пользователь настроил двухфакторную аутентификацию, клиентскому приложению может потребоваться проверить, была ли она настроена. Например, путем проверки утверждений, как `acr`в токенах.

Повторная аутентификация во время AIA

В случае, если пользователь уже аутентифицирован из-за активного сеанса SSO, этому пользователю обычно не нужно активно повторно проходить аутентификацию. Однако, если этот пользователь активно аутентифицировался более пяти минут назад, клиент все равно может запросить повторную аутентификацию, когда запрашивается некоторая AIA. Существуют следующие исключения из этого руководства:

- Действие delete_accountвсегда будет требовать от пользователя активной повторной аутентификации.
- Действие update_passwordможет потребовать от пользователя активной повторной аутентификации в соответствии с настроенной политикой

максимального возраста аутентификации пароля . Если политика не настроена, ее также можно настроить для самого требуемого действия на вкладке Требуемые действия при настройке конкретного требуемого действия. Если политика не настроена ни в одном из этих мест, по умолчанию она составляет пять минут.

- Если вы хотите использовать более короткую повторную аутентификацию, вы все равно можете использовать параметр запроса параметра, например, max_agec указанным более коротким значением или в конечном итоге prompt=login, который всегда будет требовать от пользователя активной повторной аутентификации, как описано в спецификации OIDC. Обратите внимание, что использование max_ageдля более длительного значения, чем пять минут по умолчанию (или предписанное политикой паролей), не поддерживается. max_ageВ настоящее время можно использовать только для того, чтобы сделать значение короче пяти минут по умолчанию.
- Если включена аутентификация Step-upotp и действие заключается в добавлении или удалении учетных данных, требуется аутентификация с уровнем, соответствующим заданным учетным данным. Это требование существует в случае, если у пользователя уже есть учетные данные определенного уровня. Например, если и webauthnнастроены в потоке аутентификации как аутентификаторы 2-го фактора (оба в потоке аутентификации на уровне 2) и у пользователя уже есть учетные данные 2-го фактора (otрили webauthnв этом случае), пользователю необходимо пройти аутентификацию с существующими учетными данными 2-го фактора, чтобы добавить еще одни учетные данные 2-го уровня. Таким же образом, удаление существующих учетных данных 2-го фактора (otрили webauthnв этом случае) требует аутентификации с существующими учетными данными 2-го уровня. Требование существует из соображений безопасности.

Параметризованный AIA

Некоторые AIA могут требовать отправки параметра вместе с именем действия. Например, Delete Credentialдействие может быть вызвано только AIA, и оно требует отправки параметра вместе с именем действия, которое указывает на

идентификатор удаленных учетных данных. Таким образом, URL для этого примера будет kc_action=delete_credential:ce1008ac-f811-427f-825a-c0b878d1c24b. В этом случае часть после символа двоеточия (ce1008ac-f811-427f-825a-c0b878d1c24b) содержит идентификатор учетных данных конкретного пользователя, которые должны быть удалены. Действие Delete Credentialотображает экран подтверждения, на котором пользователь может подтвердить согласие на удаление учетных данных.

> Консоль учетных записей Tuxedo SSO обычно использует это `Delete Credential`действие при удалении учетных данных 2-го фактора. Вы можете проверить Консоль учетных записей для примеров, если вы хотите использовать это действие непосредственно из своих собственных приложений. Однако лучше полагаться на Консоль учетных записей вместо управления учетными данными из своих собственных приложений.

Доступные действия

Чтобы увидеть все доступные действия, войдите в консоль администратора и перейдите в правый верхний угол, чтобы нажать Realm info→ вкладка Provider info→ Найти поставщика required-action. Но учтите, что это может быть дополнительно ограничено в зависимости от того, какие действия включены для вашей области на вкладке Требуемые действия .

Поиск пользователя

Найдите пользователя, чтобы просмотреть подробную информацию о нем, например, его группы и роли.

Предварительное условие

- Вы находитесь в той же области, где существует пользователь.

Поиск по умолчанию

Процедура

1. Нажмите Пользователи в главном меню. Отобразится эта страница Пользователи .

2. Введите полное имя, фамилию, имя или адрес электронной почты пользователя, которого вы хотите найти, в поле поиска. Поиск возвращает всех пользователей, которые соответствуют вашим критериям.

   Критерии, используемые для сопоставления пользователей, зависят от синтаксиса, используемого в строке поиска:

   a. "somevalue"→ выполняет точный поиск строки "somevalue";
   b. *somevalue*→ выполняет инфиксный поиск, аналогичный LIKE '%somevalue%'запросу к БД;
   c. somevalue*или somevalue→ выполняет префиксный поиск, аналогичный LIKE 'somevalue%'запросу к БД.

Поиск по атрибутам

Процедура

1. Нажмите Пользователи в главном меню. Отобразится эта страница Пользователи .
2. Нажмите кнопку «Поиск по умолчанию» и переключите ее на поиск по атрибутам .
3. Нажмите кнопку Выбрать атрибуты и укажите атрибуты для поиска.
4. Установите флажок Точный поиск , чтобы выполнить точное совпадение, или не устанавливайте его, чтобы использовать инфиксный поиск значений атрибутов.
5. Нажмите кнопку Поиск , чтобы выполнить поиск. Он возвращает всех пользователей, соответствующих критериям.

Поиски, выполняемые на странице **«Пользователи»,** охватывают как базу данных Tuxedo SSO, так и настроенные бэкенды федерации пользователей, такие как LDAP. Пользователи, найденные в бэкендах федерации, будут импортированы в базу данных Tuxedo SSO, если они там еще не существуют.

Дополнительные ресурсы

- Более подробную информацию о федерации пользователей см. в разделе Федерация пользователей .

Удаление пользователя

Вы можете удалить пользователя, которому больше не нужен доступ к приложениям. Если пользователь удален, профиль пользователя и данные также удаляются.

Процедура

1. Нажмите Пользователи в меню. Отобразится страница Пользователи .
2. Нажмите «Просмотреть всех пользователей» , чтобы найти пользователя, которого нужно удалить.

   Кроме того, вы можете воспользоваться строкой поиска, чтобы найти пользователя.
3. Нажмите «Удалить» в меню действий рядом с пользователем, которого вы хотите удалить, и подтвердите удаление.

Разрешение пользователям удалять учетные записи

Конечные пользователи и приложения могут удалять свои учетные записи в Account Console, если вы включите эту возможность в Admin Console. После включения этой возможности вы можете предоставить ее определенным пользователям.

Включение возможности удаления учетной записи

Эту возможность можно включить на вкладке «Требуемые действия» .

Процедура

1. Нажмите «Аутентификация» в меню.
2. Перейдите на вкладку «Требуемые действия» .
3. Выберите Включено в строке Удалить учетную запись .

Удалить учетную запись на вкладке «Необходимые действия»

Предоставление пользователю права на удаление учетной записи

Вы можете предоставить определенным пользователям роль, которая позволит удалять учетные записи.

Процедура

1. Нажмите «Пользователи» в меню.
2. Выберите пользователя.
3. Перейдите на вкладку Сопоставление ролей .
4. Нажмите кнопку Назначить роль .
5. Нажмите удалить учетную запись-аккаунт .
6. Нажмите «Назначить» .

Удалить роль учетной записи

Удаление вашего аккаунта

Получив право удалить учетную запись , вы сможете удалить свою учетную запись.

1. Войдите в консоль аккаунта.
2. В нижней части страницы «Личная информация» нажмите «Удалить учетную запись» .

Удалить страницу аккаунта

3. Введите свои учетные данные и подтвердите удаление.

Подтверждение удаления

Это действие необратимо. Все ваши данные в Tuxedo SSO будут удалены.

Выдача себя за пользователя

Администратор с соответствующими разрешениями может выдавать себя за пользователя. Например, если пользователь сталкивается с ошибкой в приложении, администратор может выдавать себя за пользователя, чтобы расследовать или дублировать проблему.

Любой пользователь с impersonationсоответствующей ролью в сфере может выдавать себя за другого пользователя.

Процедура

1. Нажмите «Пользователи» в меню.
2. Нажмите на имя пользователя, от которого хотите выдать себя.
3. В списке действий выберите пункт «Выдать себя за другого» .
    - Если администратор и пользователь находятся в одной области, то администратор выйдет из системы и автоматически войдет в систему как пользователь, от имени которого он себя выдает.
    - Если администратор и пользователь находятся в разных областях, администратор останется в системе и, кроме того, войдет в систему как пользователь в области этого пользователя.

В обоих случаях отображается консоль учетной записи выдаваемого пользователя.

Дополнительные ресурсы

- Дополнительную информацию о назначении прав администратора см. в главе «Управление доступом к консоли администратора» .

Включение reCAPTCHA

Для защиты регистрации от ботов Tuxedo SSO имеет интеграцию с Google reCAPTCHA (см. Настройка Google reCAPTCHA) и reCAPTCHA Enterprise (см. Настройка Google reCAPTCHA Enterprise). Тема по умолчанию (register.ftl) поддерживает как v2 (видимую, основанную на флажках), так и v3 (основанную на баллах, невидимую) reCAPTCHA (см. Выберите подходящий тип ключа reCAPTCHA).

Настройка Google reCAPTCHA

1. Введите следующий URL-адрес в браузере:

   https://www.google.com/recaptcha/admin/create

2. Создайте reCAPTCHA и выберите Challenge v2 (видимый флажок) или Score-based, v3 (невидимый), чтобы получить ключ и секрет сайта reCAPTCHA. Запишите их для будущего использования в этой процедуре.

   По умолчанию работает localhost. Домен указывать не нужно.
3. Перейдите в консоль администратора Tuxedo SSO.
4. Нажмите «Аутентификация» в меню.
5. Перейдите на вкладку Потоки .
6. Выберите из списка пункт «Регистрация» .
7. Установите требование reCAPTCHA на Required . Это включает reCAPTCHA.
8. Нажмите на значок шестеренки в строке reCAPTCHA .

   Конфигурация reCAPTCHA

   a. Введите ключ сайта reCAPTCHA , сгенерированный на сайте Google reCAPTCHA.
   b. Введите секрет reCAPTCHA , сгенерированный на сайте Google reCAPTCHA.
   c. Включите reCAPTCHA v3 в соответствии с типом ключа вашего сайта: включите для reCAPTCHA на основе очков (v3), выключите для reCAPTCHA с вызовом (v2).
   d. (Необязательно) Переключить Использовать recaptcha.net для использования www.recatcha.netвместо www.google.comдомена для файлов cookie. См. reCAPTCHA faq для получения дополнительной информации.
9. Разрешите Google использовать страницу регистрации в качестве iframe.

В Tuxedo SSO веб-сайты не могут включать диалоговое окно страницы входа в iframe. Это ограничение необходимо для предотвращения атак clickjacking. Вам необходимо изменить заголовки HTTP-ответов по умолчанию, установленные в Tuxedo SSO.

a. Нажмите «Настройки области» в меню.
b. Перейдите на вкладку «Защита» .
c. Введите https://www.google.comв поле заголовка X-Frame-Options (или https//www.recaptcha.net, если вы включили Use recaptcha.net).
d. Введите значение https://www.google.comв поле заголовка Content-Security-Policy (или https//www.recaptcha.net, если вы включили опцию Use recaptcha.net).

Настройка Google reCAPTCHA Enterprise

1. Введите следующий URL-адрес в браузере:

   https://developers.google.com/recaptcha/

2. Создайте ключ для платформы "Веб-сайт" и выберите нужный тип ключа. Оставьте значения по умолчанию для v3 reCAPTCHA (невидимый) или переключите флажок Использовать вызов для v2 reCAPTCHA (видимый). Запишите ключ сайта для будущего использования в этой процедуре.

   По умолчанию работает localhost. Домен указывать не нужно.

3. В проекте Google Cloud перейдите в раздел «Учетные данные» и создайте ключ API.

   Для большей безопасности нажмите **«Изменить ключ API»** и добавьте ограничение API, чтобы ограничить ключ только **API reCAPTCHA Enterprise** .

4. Перейдите в консоль администратора Tuxedo SSO.
5. Нажмите «Аутентификация» в меню.
6. Перейдите на вкладку Потоки .
7. Дублируйте поток «регистрации».
8. Свяжите новый поток с потоком регистрации .
9. Отредактируйте новый поток:

a. Удалить шаг reCAPTCHA .

b. Добавьте шаг reCAPTCHA Enterprise как подшаг «регистрационной формы» (первый шаг процесса).

10. Установите требование reCAPTCHA Enterprise на «Обязательно» .

11. Нажмите на значок шестеренки ⚙ в строке reCAPTCHA Enterprise .

Конфигурация reCAPTCHA Enterprise

a. Введите идентификатор проекта Recaptcha вашего проекта консоли Google Cloud.

b. Введите ключ сайта Recaptcha, сгенерированный в начале процедуры.

c. Введите API-ключ Recaptcha, сгенерированный в начале процедуры.

d. Включите reCAPTCHA v3 в соответствии с типом ключа вашего сайта: включите для reCAPTCHA на основе очков (v3), выключите для reCAPTCHA с вызовом (v2).

e. (Необязательно) Настройте Min. Score Threshold по своему усмотрению. Установите его на минимальный балл от 0,0 до 1,0, который пользователь должен получить в reCAPTCHA, чтобы ему разрешили зарегистрироваться. См. интерпретацию баллов .

f. (Необязательно) Переключить Использовать recaptcha.net для использования www.recatcha.netвместо www.google.comдомена для файлов cookie. См. reCAPTCHA faq для получения дополнительной информации.

12. Разрешите Google использовать страницу регистрации как iframe. Подробную процедуру см. в последних шагах Настройка Google reCAPTCHA .

Дополнительные ресурсы

- Дополнительную информацию о расширении и создании тем см. в Руководстве разработчика сервера .

Персональные данные, собираемые Tuxedo SSO

По умолчанию Tuxedo SSO собирает следующие данные:

- Основные данные профиля пользователя, такие как адрес электронной почты, имя и фамилия пользователя.
- Основные данные профиля пользователя, используемые для учетных записей в социальных сетях, и ссылки на учетную запись в социальных сетях при использовании входа через социальную сеть.
- Информация об устройстве, собираемая в целях аудита и безопасности, такая как IP-адрес, имя операционной системы и имя браузера.

Информация, собранная в Tuxedo SSO, легко настраивается. При выполнении настроек применяются следующие рекомендации:

- Формы регистрации и учетной записи могут содержать настраиваемые поля, такие как день рождения, пол и национальность. Администратор может настроить Tuxedo SSO для извлечения данных из социального провайдера или провайдера хранилища пользователей, такого как LDAP.
- Tuxedo SSO собирает учетные данные пользователя, такие как пароль, коды OTP и открытые ключи WebAuthn. Эта информация шифруется и сохраняется в базе данных, поэтому она не видна администраторам Tuxedo SSO. Каждый тип учетных данных может включать неконфиденциальные метаданные, которые видны администраторам, такие как алгоритм, используемый для хеширования пароля, и количество итераций хеширования, используемых для хеширования пароля.
- При включенных службах авторизации и поддержке UMA Tuxedo SSO может хранить информацию о некоторых объектах, владельцем которых является конкретный пользователь.

Управление сеансами пользователей

Когда пользователи входят в области, Tuxedo SSO поддерживает сеанс пользователя для каждого пользователя и запоминает каждого клиента, посещенного пользователем в течение сеанса. Администраторы областей могут выполнять несколько действий в каждом сеансе пользователя:

- Просмотр статистики входов в систему.
- Просмотр активных пользователей и того, где они вошли в систему.
- Выйти из сеанса пользователя.
- Отозвать токены.
- Настройте тайм-ауты токенов.
- Настройте тайм-ауты сеанса.

Администрирование сессий

Чтобы просмотреть общее представление активных клиентов и сеансов в Tuxedo SSO, выберите в меню пункт «Сеансы» .

Сессии

Выход из всех активных сеансов

Вы можете выйти из системы всех пользователей в области. В списке действий выберите Выйти из всех активных сеансов . Все файлы cookie SSO становятся недействительными. Tuxedo SSO уведомляет клиентов с помощью клиентского адаптера Tuxedo SSO OIDC о событии выхода из системы. Клиенты, запрашивающие аутентификацию в активных сеансах браузера, должны войти в систему снова. Такие типы клиентов, как SAML, не получают запрос на выход из системы по обратному каналу.

> Нажатие кнопки **«Выйти из всех активных сеансов»** не отменяет выдающиеся токены доступа. Выдающиеся токены должны истекать естественным образом. Для клиентов, использующих клиентский адаптер Tuxedo SSO OIDC, вы можете применить политику отзыва, чтобы отозвать токен, но это не работает для других адаптеров.

Просмотр клиентских сессий

Процедура

1. Нажмите «Клиенты» в меню.
2. Нажмите на имя клиента, чтобы просмотреть сеансы этого клиента.
3. Нажмите вкладку Сеансы .

   Клиентские сессии

Просмотр сеансов пользователей

Процедура

1. Нажмите «Пользователи» в меню.
2. Щелкните пользователя, чтобы просмотреть его сеансы.
3. Нажмите вкладку Сеансы .

   Пользовательские сеансы

Отмена активных сессий

Если ваша система скомпрометирована, вы можете отозвать все активные сеансы и токены доступа.

Процедура

1. Нажмите «Сеансы» в меню.
2. В списке действий выберите Отзыв .

   Отзыв

3. Укажите время и дату, когда сеансы или токены, выпущенные до этого времени и даты, будут недействительны с помощью этой консоли.
   - Нажмите Установить сейчас , чтобы установить политику на текущее время и дату.

- Нажмите «Push» , чтобы передать эту политику отзыва любому зарегистрированному клиенту OIDC с клиентским адаптером Tuxedo SSO OIDC.

Тайм-ауты сеанса и токена

Tuxedo SSO включает в себя управление тайм-аутами сеанса, cookie-файлов и токенов через вкладки «Сеансы» и «Токены» в меню настроек Realm .

Вкладка «Сеансы»

Конфигурация	Описание
Сеанс SSO в режиме ожидания	Эта настройка предназначена только для клиентов OIDC. Если пользователь неактивен дольше этого тайм-аута, сеанс пользователя становится недействительным. Это значение тайм-аута сбрасывается, когда клиенты запрашивают аутентификацию или отправляют запрос на обновление токена. Tuxedo SSO добавляет временной интервал к тайм-ауту простоя, прежде чем аннулирование сеанса вступит в силу. См. примечание далее в этом разделе.
Максимальный сеанс SSO	Максимальное время до истечения сеанса пользователя.
Сеанс SSO в режиме ожидания Запомнить меня	Эта настройка похожа на стандартную конфигурацию простоя сеанса SSO, но относится к входам с включенной функцией **«Запомнить меня»** . Пользователи могут указать более длительные тайм-ауты простоя сеанса, нажав **«Запомнить меня»** при входе в систему. Эта настройка является необязательной конфигурацией и, если ее значение не больше нуля, она использует тот же тайм-аут простоя, что и конфигурация простоя сеанса SSO.
SSO сессия Макс Запомнить меня	Эта настройка похожа на стандартную настройку SSO Session Max, но относится только к входам **Remember Me** . Пользователи могут указать более длительные сеансы, нажав **Remember Me** при входе. Эта настройка является необязательной конфигурацией и, если ее значение не больше нуля, она использует ту же продолжительность сеанса, что и конфигурация SSO Session Max.
Клиентский сеанс в режиме ожидания	Время ожидания бездействия для клиентского сеанса. Если пользователь неактивен дольше этого времени ожидания, клиентский сеанс становится недействительным, а запросы токена обновления увеличивают время ожидания бездействия. Этот параметр никогда не влияет на общий сеанс

Конфигурация	Описание
	пользователя SSO, который является уникальным. Обратите внимание, что сеанс пользователя SSO является родительским для нуля или более клиентских сеансов, один клиентский сеанс создается для каждого отдельного клиентского приложения, в которое входит пользователь. Это значение должно указывать более короткое время ожидания бездействия, чем **SSO Session Idle** . Пользователи могут переопределить его для отдельных клиентов на вкладке клиента **Дополнительные параметры** . Этот параметр является необязательной конфигурацией и, если установлен на ноль, использует то же время ожидания бездействия в конфигурации SSO Session Idle.
Макс. сеанс клиента	Максимальное время для сеанса клиента и до истечения срока действия токена обновления и его аннулирования. Как и в предыдущем варианте, этот параметр никогда не влияет на сеанс пользователя SSO и должен указывать более короткое значение, чем **SSO Session Max** . Пользователи могут переопределить его для отдельных клиентов на вкладке клиента **Advanced Settings** . Этот параметр является необязательной конфигурацией и, если установлен на ноль, использует тот же максимальный тайм-аут в конфигурации SSO Session Max.
Оффлайн сеанс в режиме ожидания	Эта настройка предназначена для офлайн-доступа . Время, в течение которого сеанс остается бездействующим, прежде чем Tuxedo SSO отзовет свой офлайн-токен. Tuxedo SSO добавляет временной интервал к тайм-ауту бездействия, прежде чем аннулирование сеанса вступит в силу. См. примечание далее в этом разделе.
Максимальное количество сеансов в автономном режиме ограничено	Этот параметр предназначен для офлайн-доступа . Если этот флаг **включен** , Offline Session Max может контролировать максимальное время, в течение которого офлайн-токен остается активным, независимо от активности пользователя. Если флаг **отключен** , офлайн-сеансы никогда не истекают по сроку действия, только по причине бездействия. После активации этого параметра можно настроить Offline Session Max (глобальный параметр на уровне области) и **Client Offline Session Max** (конкретный параметр уровня клиента на вкладке **Advanced Settings**).
Максимальное количество сеансов в автономном режиме	Эта настройка предназначена для офлайн-доступа и является максимальным временем, по истечении которого Tuxedo SSO отзовет соответствующий офлайн-токен. Эта опция контролирует максимальное время, в течение которого офлайн-токен остается активным, независимо от активности пользователя.
Истекло время	Общее время, которое должен занять вход в систему. Если аутентификация

Конфигурация	Описание
ожидания входа	занимает больше этого времени, пользователь должен начать процесс аутентификации заново.
Истекло время ожидания действия входа	Максимальное время, которое пользователи могут провести на любой странице в процессе аутентификации.

Вкладка «Токены»

Конфигурация	Описание
Алгоритм подписи по умолчанию	Алгоритм по умолчанию, используемый для назначения токенов для области.
Отозвать токен обновления	При **Enabled** Tuxedo SSO отзывает токены обновления и выпускает другой токен, который должен использовать клиент. Это действие применяется к клиентам OIDC, выполняющим поток токенов обновления.
Срок действия токена доступа	Когда Tuxedo SSO создает токен доступа OIDC, это значение управляет сроком действия токена.
Срок действия токена доступа для неявного потока	С Implicit Flow Tuxedo SSO не предоставляет токен обновления. Для токенов доступа, созданных Implicit Flow, существует отдельный тайм-аут.
Время ожидания входа клиента	Максимальное время, прежде чем клиенты должны завершить поток кода авторизации в OIDC.
Продолжительность действия, инициированного пользователем	Максимальное время до истечения срока действия разрешения пользователя на действие. Сохраняйте это значение коротким, поскольку пользователи обычно быстро реагируют на самостоятельно созданные действия.
Срок действия действия, инициированного администратором по умолчанию	Максимальное время до истечения срока действия разрешения на действие, отправленного пользователю администратором. Сохраняйте это значение большим, чтобы администраторы могли отправлять электронные письма пользователям, находящимся в автономном режиме. Администратор может переопределить тайм-аут по умолчанию перед выдачей токена.

Конфигурация	Описание
Проверка электронной почты	Задает независимый тайм-аут для проверки электронной почты.
Проверка электронной почты учетной записи IdP	Задает независимый тайм-аут для проверки адреса электронной почты учетной записи IdP.
Забыли пароль	Задает независимый тайм-аут для забытого пароля.
Выполнять действия	Задает независимый тайм-аут для выполнения действий.

Следующая логика применяется только в том случае, если постоянные сеансы пользователей неактивны:

Для тайм-аутов простоя существует двухминутное окно времени, в течение которого сеанс активен. Например, если вы установили тайм-аут на 30 минут, то до истечения сеанса пройдет 32 минуты.

Это действие необходимо для некоторых сценариев в кластерных и кросс-центровых средах, где токен обновляется на одном узле кластера незадолго до истечения срока действия, а другие узлы кластера ошибочно считают сеанс истекшим, поскольку они еще не получили сообщение об успешном обновлении от обновляющего узла.

Оффлайн доступ

Во время входа в систему офлайн-доступа клиентское приложение запрашивает офлайн-токен вместо токена обновления. Клиентское приложение сохраняет этот офлайн-токен и может использовать его для будущих входов, если пользователь выходит из системы. Это действие полезно, если вашему приложению необходимо выполнять офлайн-действия от имени пользователя, даже когда пользователь не в сети. Например, регулярное резервное копирование данных.

Клиентское приложение отвечает за сохранение автономного токена в хранилище и его последующее использование для получения новых токенов доступа с сервера Tuxedo SSO.

Разница между токеном обновления и офлайн-токеном заключается в том, что офлайн-токен никогда не истекает и не подлежит SSO Session Idletайм-ауту и SSO Session Maxсроку службы. Офлайн-токен действителен после выхода пользователя из системы. Вы должны использовать офлайн-токен для действия токена обновления не реже одного раза в тридцать дней или для значения Offline Session Idle .

Если вы включите Offline Session Max Limited , офлайн-токены истекают через 60 дней, даже если вы используете офлайн-токен для действия обновления токена. Вы можете изменить это значение, Offline Session Max , в консоли администратора.

При использовании автономного доступа время простоя клиента и максимальные тайм-ауты можно переопределить на уровне клиента . Параметры Client Offline Session Idle и Client Offline Session Max на вкладке Дополнительные параметры клиента позволяют вам иметь более короткие тайм-ауты в автономном режиме для определенного приложения. Обратите внимание, что значения сеанса клиента также управляют истечением срока действия маркера обновления, но они никогда не влияют на глобальный сеанс единого входа автономного пользователя. Параметр Client Offline Session Max оценивается в клиенте, только если Offline Session Max Limited включен на уровне области.

Если вы включите опцию Revoke Refresh Token , вы сможете использовать каждый offline token только один раз. После обновления вы должны сохранить новый offline token из ответа на обновление вместо предыдущего.

Пользователи могут просматривать и отзывать офлайн-токены, которые Tuxedo SSO предоставляет им в консоли учетных записей пользователей . Администраторы могут отзывать офлайн-токены для отдельных пользователей в консоли администратора на Consentsвкладке. Администраторы могут просматривать все офлайн-токены, выпущенные на Offline Accessвкладке каждого клиента. Администраторы могут отзывать офлайн-токены, установив политику отзыва .

Для выпуска офлайн-токена пользователи должны иметь сопоставление ролей для offline_accessроли уровня области. Клиенты также должны иметь эту роль в

своей области. Клиенты должны добавить offline_accessклиентскую область в качестве Optional client scopeроли, что делается по умолчанию.

Клиенты могут запросить автономный токен, добавив параметр scope=offline_accessпри отправке своего запроса на авторизацию в Tuxedo SSO. Клиентский адаптер OIDC Tuxedo SSO автоматически добавляет этот параметр, когда вы используете его для доступа к защищенному URL вашего приложения (например, http://localhost:8080/customer-portal/secured?scope=offline_access). Учетные записи Direct Access Grant и Service поддерживают автономные токены, если вы включаете их scope=offline_accessв тело запроса аутентификации.

Tuxedo SSO ограничит свой внутренний кэш для сеансов офлайн-пользователей и офлайн-клиентов до 10000 записей по умолчанию, что снизит общее использование памяти для офлайн-сеансов. Элементы, которые вытесняются из памяти, будут загружаться по требованию из базы данных при необходимости. Чтобы задать разные размеры для кэшей, отредактируйте файл конфигурации кэша Tuxedo SSO, чтобы задать <memory max-count="..."/>для этих кэшей.

Если вы отключили функцию persistent-user-sessions, можно снизить требования к памяти с помощью параметра конфигурации, который сокращает срок службы импортированных офлайн-сессий. Такие сессии будут вытеснены из кэшей Infinispan после указанного срока службы, но по-прежнему будут доступны в базе данных. Это снизит потребление памяти, особенно для развертываний с большим количеством офлайн-сессий.

Чтобы указать переопределение срока действия для сеансов пользователей, находящихся в автономном режиме, запустите сервер Tuxedo SSO со следующим параметром:

--spi-user-sessions-infinispan-offline-session-cache-entry-lifespan-override=<lifespan-in-seconds>

Аналогично для сеансов офлайн-клиентов:

--spi-user-sessions-infinispan-offline-client-session-cache-entry-lifespan-override=<lifespan-in-seconds>

Краткосрочные сеансы

Вы можете проводить временные сеансы в Tuxedo SSO. При использовании временных сеансов Tuxedo SSO не создает сеанс пользователя после успешной аутентификации. Tuxedo SSO создает временный временный сеанс для области действия текущего запроса, который успешно аутентифицирует пользователя. Tuxedo SSO может запускать сопоставители протоколов , используя временные сеансы после аутентификации.

sidИ токенов session_stateобычно пусты, когда токен выдается с временными сеансами. Поэтому во время временных сеансов клиентское приложение не может обновить токены или проверить определенный сеанс. Иногда эти действия не нужны, поэтому вы можете избежать дополнительного использования ресурсов сохраняющимися сеансами пользователей. Этот сеанс экономит ресурсы производительности, памяти и сетевой коммуникации (в кластерных и кросс-центровых средах).

В данный момент временные сеансы автоматически используются только во время аутентификации учетной записи службы с отключенным обновлением токена. Обратите внимание, что обновление токена автоматически отключено во время аутентификации учетной записи службы, если оно явно не включено клиентским переключателем Use refresh tokens for client credentials grant.

Назначение разрешений с использованием ролей и групп

Роли и группы имеют схожую цель — предоставить пользователям доступ и разрешения на использование приложений. Группы — это набор пользователей, к которым вы применяете роли и атрибуты. Роли определяют разрешения конкретных приложений и контроль доступа.

Роль обычно применяется к одному типу пользователя. Например, организация может включать роли admin, user, managerи employee. Приложение может назначать доступ и разрешения роли, а затем назначать эту роль нескольким пользователям, чтобы пользователи имели одинаковый доступ и разрешения.

Например, в консоли администратора есть роли, которые дают пользователям разрешение на доступ к различным частям консоли администратора.

Существует глобальное пространство имен для ролей, и каждый клиент также имеет свое собственное выделенное пространство имен, где могут быть определены роли.

Создание роли области

Роли уровня Realm — это пространство имен для определения ваших ролей. Чтобы увидеть список ролей, щелкните Realm Roles в меню.

Процедура

1. Нажмите Создать роль .
2. Введите имя роли .
3. Введите описание .
4. Нажмите «Сохранить ».

Добавить роль

Поле описания можно локализовать, указав переменную подстановки со ${var-name}строками. Локализованное значение настраивается для вашей темы в файлах свойств тем. Подробнее см. в руководстве разработчика сервера .

Роли клиентов

Роли клиента — это пространства имен, выделенные для клиентов. Каждый клиент получает свое собственное пространство имен. Роли клиента управляются на вкладке Роли для каждого клиента. Вы взаимодействуете с этим пользовательским интерфейсом так же, как и для ролей уровня области.

Преобразование роли в составную роль

Любая роль уровня области или клиента может стать составной ролью . Составная роль — это роль, с которой связана одна или несколько дополнительных ролей. Когда составная роль сопоставляется пользователю, пользователь получает роли, связанные с составной ролью. Это наследование рекурсивно, поэтому пользователи также наследуют любую составную роль составных ролей. Однако мы рекомендуем не злоупотреблять составными ролями.

Процедура

1. Нажмите «Роли области» в меню.
2. Щелкните роль, которую вы хотите преобразовать.
3. В списке действий выберите Добавить связанные роли .

Композитная роль

Пользовательский интерфейс выбора ролей отображается на странице, и вы можете связать роли уровня области и уровня клиента с создаваемой вами составной ролью.

В этом примере роль уровня сферы сотрудника связана с составной ролью разработчика . Любой пользователь с ролью разработчика также наследует роль сотрудника .

> При создании токенов и утверждений SAML любой композит также имеет свои связанные роли, добавленные к утверждениям и утверждениям ответа аутентификации, отправляемого обратно клиенту.

Назначение сопоставлений ролей

Вы можете назначить сопоставления ролей пользователю через вкладку «Сопоставления ролей» для этого пользователя.

Процедура

1. Нажмите «Пользователи» в меню.
2. Щелкните пользователя, для которого вы хотите выполнить сопоставление ролей.

3. Перейдите на вкладку Сопоставление ролей .
4. Нажмите Назначить роль .
5. Выберите в диалоговом окне роль, которую вы хотите назначить пользователю.
6. Нажмите «Назначить» .

Сопоставление ролей

В предыдущем примере мы назначаем пользователю составную роль разработчика . Эта роль была создана в теме Составные роли .

Эффективное распределение ролей

Когда назначается роль разработчика , роль сотрудника , связанная с композитом разработчика, отображается с Inherited "True". Унаследованные роли — это роли, явно назначенные пользователям, и роли, которые унаследованы от композитов.

Использование ролей по умолчанию

Используйте роли по умолчанию для автоматического назначения сопоставлений ролей пользователей при создании или импорте пользователя через Identity Brokering .

Процедура

1. Нажмите «Настройки области» в меню.
2. Нажмите вкладку Регистрация пользователя .

   Роли по умолчанию

На этом снимке экрана показано, что некоторые роли по умолчанию уже существуют.

Сопоставление областей действия ролей

При создании токена доступа OIDC или утверждения SAML сопоставления ролей пользователей становятся утверждениями в токене или утверждении. Приложения используют эти утверждения для принятия решений о доступе к ресурсам, контролируемым приложением. Tuxedo SSO цифровым образом подписывает токены доступа, и приложения повторно используют их для вызова удаленно защищенных служб REST. Однако эти токены имеют связанный риск. Злоумышленник может получить эти токены и использовать их разрешения для компрометации ваших сетей. Чтобы предотвратить эту ситуацию, используйте сопоставления областей действия ролей .

Сопоставления областей ролей ограничивают роли, объявленные внутри токена доступа. Когда клиент запрашивает аутентификацию пользователя, токен доступа, который он получает, содержит только сопоставления ролей, которые явно указаны для области клиента. Результатом является то, что вы ограничиваете разрешения каждого отдельного токена доступа вместо того, чтобы предоставить клиенту доступ ко всем разрешениям пользователей.

По умолчанию каждый клиент получает все сопоставления ролей пользователя. Вы можете просмотреть сопоставления ролей для клиента.

Процедура

1. Нажмите «Клиенты» в меню.
2. Нажмите на имя клиента, чтобы перейти к подробностям.
3. Перейдите на вкладку «Области действия клиента» .
4. Нажмите ссылку в строке с выделенной областью и картографами для этого клиента.
5. Перейдите на вкладку «Область действия» .

Полный объем

По умолчанию эффективные роли областей действия — это каждая объявленная роль в области. Чтобы изменить это поведение по умолчанию, переключите Full Scope Allowed в положение OFF и объявите конкретные роли, которые вы хотите в каждом клиенте. Вы также можете использовать клиентские области действия ,

чтобы определить те же сопоставления областей действия ролей для набора клиентов.

Частичная сфера применения

Группы

Группы в Tuxedo SSO управляют общим набором атрибутов и сопоставлений ролей для каждого пользователя. Пользователи могут быть членами любого количества групп и наследовать атрибуты и сопоставления ролей, назначенные каждой группе.

Для управления группами нажмите «Группы» в меню.

Группы

Группы иерархичны. Группа может иметь несколько подгрупп, но группа может иметь только одного родителя. Подгруппы наследуют атрибуты и сопоставления ролей от своего родителя. Пользователи также наследуют атрибуты и сопоставления ролей от своего родителя.

Если у вас есть родительская группа и дочерняя группа, а также пользователь, который принадлежит только дочерней группе, пользователь в дочерней группе наследует атрибуты и сопоставления ролей как родительской группы, так и дочерней группы.

Иерархия группы иногда представлена с помощью группового пути. Путь — это полный список имен, представляющих иерархию определенной группы, сверху вниз и разделенных косыми чертами /(аналогично файлам в файловой системе). Например, путь может быть /top/level1/level2, что означает, что topэто группа верхнего уровня и является родительской для level1, которая, в свою очередь, является родительской для level2. Этот путь однозначно представляет иерархию для группы level2.

По историческим причинам Tuxedo SSO не экранирует слеши в самом имени группы. Поэтому группа, названная level1/groupниже, toриспользует путь /top/level1/group, что вводит в заблуждение. Tuxedo SSO можно запустить с

опцией --spi-group-jpa-escape-slashes-in-group-pathto true, а затем слеши в имени экранируются символом ~. Символ экранирования обозначает, что слеш является частью имени и не имеет иерархического значения. Предыдущий пример пути был бы /top/level1~/groupпри экранировании.

bin/kc.[sh|bat] start --spi-group-jpa-escape-slashes-in-group-path=true

Следующий пример включает группу продаж верхнего уровня и дочернюю подгруппу «Северная Америка» .

Чтобы добавить группу:

1. Нажмите на группу.
2. Нажмите Создать группу .
3. Введите название группы.
4. Нажмите «Создать» .
5. Нажмите на название группы.

   Откроется страница управления группой.

   Группа

Атрибуты и сопоставления ролей, которые вы определяете, наследуются группами и пользователями, являющимися членами группы.

Чтобы добавить пользователя в группу:

1. Нажмите «Пользователи» в меню.
2. Щелкните пользователя, которому вы хотите выполнить сопоставление ролей. Если пользователь не отображается, щелкните Просмотреть всех пользователей .
3. Нажмите Группы .

   Группы пользователей
4. Нажмите Присоединиться к группе .
5. Выберите группу в диалоговом окне.

6. Выберите группу из дерева доступных групп .
7. Нажмите «Присоединиться» .

Чтобы удалить группу из пользователя:

1. Нажмите «Пользователи» в меню.
2. Щелкните пользователя, которого необходимо удалить из группы.
3. Нажмите «Выйти» в строке таблицы группы.

В этом примере пользователь jimlincoln находится в группе North America . Вы можете увидеть jimlincoln, отображаемый на вкладке Members для группы.

Членство в группе

Группы в сравнении с ролями

Группы и роли имеют некоторые сходства и различия. В Tuxedo SSO группы представляют собой набор пользователей, к которым вы применяете роли и атрибуты. Роли определяют типы пользователей, а приложения назначают разрешения и контроль доступа ролям.

Составные роли похожи на группы, поскольку они предоставляют ту же функциональность. Разница между ними концептуальная. Составные роли применяют модель разрешений к набору служб и приложений. Используйте составные роли для управления приложениями и службами.

Группы фокусируются на коллекциях пользователей и их ролях в организации. Используйте группы для управления пользователями.

Использование групп по умолчанию

Чтобы автоматически назначать членство в группе всем пользователям, созданным или импортированным через Identity Brokering , используются группы по умолчанию.

1. Нажмите «Настройки области» в меню.

2. Нажмите вкладку Регистрация пользователя .
3. Перейдите на вкладку Группы по умолчанию .

   Группы по умолчанию

На этом снимке экрана показано, что некоторые группы по умолчанию уже существуют.

Настройка аутентификации

В этой главе рассматриваются несколько тем аутентификации. Эти темы включают:

- Внедрение строгих политик использования паролей и одноразовых паролей (OTP).
- Управление различными типами учетных данных.
- Вход с помощью Kerberos.
- Отключение и включение встроенных типов учетных данных.

Политика паролей

Когда Tuxedo SSO создает область, он не связывает политики паролей с областью. Вы можете задать простой пароль без ограничений по длине, безопасности или сложности. Простые пароли неприемлемы в производственных средах. Tuxedo SSO имеет набор политик паролей, доступных через консоль администратора.

Процедура

1. Нажмите «Аутентификация» в меню.
2. Перейдите на вкладку «Политики» .
3. Выберите политику для добавления в раскрывающемся списке Добавить политику .
4. Введите значение, которое применяется к выбранной политике.

5. Нажмите «Сохранить ».

Политика паролей

После сохранения политики Tuxedo SSO применяет ее для новых пользователей.

> Новая политика не будет эффективна для существующих пользователей. Поэтому убедитесь, что вы установили политику паролей с самого начала создания области или добавьте «Обновить пароль» для существующих пользователей или используйте «Истекает пароль», чтобы убедиться, что пользователи обновляют свои пароли в течение следующих «N» дней, что фактически подстроится под новые политики паролей.

Типы политики паролей

HashAlgorithm
Пароли не хранятся в открытом виде. Перед сохранением или проверкой Tuxedo SSO хэширует пароли, используя стандартные алгоритмы хэширования.

Поддерживаемые алгоритмы хеширования паролей включают:

- argon2:: Argon2 (по умолчанию для развертываний, не соответствующих FIPS)
- pbkdf2-sha512:: PBKDF2 с SHA512 (по умолчанию для развертываний FIPS)
- pbkdf2-sha256:: PBKDF2 с SHA256
- pbkdf2:: PBKDF2 с SHA1 (устарело)

Настоятельно рекомендуется использовать Argon2, когда это возможно, поскольку он предъявляет значительно меньшие требования к процессору по сравнению с PBKDF2, но в то же время является более безопасным.

Алгоритм хеширования паролей по умолчанию для сервера можно настроить с помощью --spi-password-hashing-provider-default=<algorithm>.

Чтобы предотвратить чрезмерное использование памяти и ЦП, параллельное вычисление хэшей Argon2 по умолчанию ограничено количеством ядер, доступных JVM. Для настройки поставщика хэширования Argon2 используйте его параметры поставщика.

Информацию о том, как добавить собственный алгоритм хеширования, см. в Руководстве разработчика сервера .

Если изменить алгоритм хеширования, хэши паролей в хранилище не изменятся до тех пор, пока пользователь не войдет в систему.

Итерации хеширования
Указывает количество раз, которое Tuxedo SSO хеширует пароли перед сохранением или проверкой. Значение по умолчанию -1, которое использует интервалы хеширования по умолчанию для выбранного алгоритма хеширования:

- аргон2:: 5
- pbkdf2-sha512:: 210,000
- pbkdf2-sha256:: 600,000
- pbkdf2:: 1,300,000

В большинстве случаев итерации хеширования не следует изменять с рекомендуемых значений по умолчанию. Более низкие значения итераций обеспечивают недостаточную безопасность, в то время как более высокие значения приводят к более высоким требованиям к мощности ЦП.

Цифры
Количество цифр, необходимое в строке пароля.

Строчные буквы
Количество строчных букв, необходимое в строке пароля.

Заглавные буквы
Количество заглавных букв, необходимое в строке пароля.

Специальные символы
Количество специальных символов, необходимых в строке пароля.

Не имя пользователя
Пароль не может совпадать с именем пользователя.

Не электронная почта
Пароль не может совпадать с адресом электронной почты пользователя.

Регулярное выражение
Пароль должен соответствовать одному или нескольким определенным шаблонам регулярных выражений Java. Синтаксис этих выражений см . в документации по регулярным выражениям Java .

Срок действия пароля истек
Количество дней, в течение которых пароль действителен. Когда количество дней истекает, пользователь должен сменить свой пароль.

Недавно не использовался
Пароль не может быть уже использован пользователем. Tuxedo SSO хранит историю использованных паролей. Количество сохраненных старых паролей настраивается в Tuxedo SSO.

Недавно не использовался (в днях)
Пароль не может быть повторно использован в течение настроенного периода времени (в днях). Если новый пароль был последний раз установлен в течение этого периода, пользователь будет вынужден предоставить другой.

Черный список паролей
Пароль не должен находиться в файле черного списка.

- Файлы черного списка — это простые текстовые файлы UTF-8 с окончаниями строк Unix. Каждая строка представляет собой пароль из черного списка.
- Tuxedo SSO сравнивает пароли без учета регистра.
- Значение файла черного списка должно быть именем файла черного списка, например, 100k_passwords.txt.
- Файлы черного списка разрешаются ${kc.home.dir}/data/password-blacklists/по умолчанию. Настройте этот путь с помощью:
    - Системное Tuxedo SSO.password.blacklists.pathсвойство.

- Свойство конфигурации blacklistsPathполитики passwordBlacklistSPI. Чтобы настроить папку черного списка с помощью CLI, используйте --spi-password-policy-password-blacklist-blacklists-path=/path/to/blacklistsFolder.

Примечание о ложных срабатываниях

Текущая реализация использует BloomFilter для быстрой и эффективной проверки наличия паролей, например, на наличие определенного пароля в черном списке, с возможностью ложных срабатываний.

- 0.01%По умолчанию используется вероятность ложного срабатывания .
- Чтобы изменить вероятность ложного срабатывания с помощью конфигурации CLI, используйте --spi-password-policy-password-blacklist-false-positive-probability=0.00001.

Максимальный возраст аутентификации
Указывает максимальный возраст аутентификации пользователя в секундах, с которым пользователь может обновить пароль без повторной аутентификации. Значение 0указывает, что пользователь должен всегда повторно аутентифицироваться с текущим паролем, прежде чем он сможет обновить пароль. Дополнительные сведения об этой политике см. в разделе AIA .

> Максимальный возраст аутентификации также можно настроить при настройке требуемого действия **«Обновить пароль»** на вкладке **«Требуемые действия»** в консоли администратора. Лучшим выбором будет использование требуемого действия для настройки, поскольку политика паролей *«Максимальный возраст аутентификации»* может быть устаревшей/удалена в будущем.

Политики одноразовых паролей (OTP)

Tuxedo SSO имеет несколько политик по настройке генератора одноразовых паролей FreeOTP или Google Authenticator.

Процедура

1. Нажмите «Аутентификация» в меню.
2. Перейдите на вкладку «Политика» .

3. Перейдите на вкладку «Политика OTP» .

Политика OTP

Tuxedo SSO генерирует QR-код на странице настройки OTP на основе информации, настроенной на вкладке Политика OTP . FreeOTP и Google Authenticator сканируют QR-код при настройке OTP.

Одноразовые пароли, основанные на времени или счетчике

Алгоритмы, доступные в Tuxedo SSO для генераторов OTP, основаны на времени и счетчиках.

С Time-Based One Time Passwords (TOTP) генератор токенов будет хэшировать текущее время и общий секрет. Сервер проверяет OTP, сравнивая хэши в течение определенного промежутка времени с представленным значением. TOTP действительны в течение короткого промежутка времени.

С помощью Counter-Based One Time Passwords (HOTP) Tuxedo SSO использует общий счетчик, а не текущее время. Сервер Tuxedo SSO увеличивает счетчик с каждым успешным входом OTP. Действительные OTP изменяются после успешного входа.

TOTP более безопасен, чем HOTP, поскольку сопоставляемый OTP действителен в течение короткого промежутка времени, в то время как OTP для HOTP действителен в течение неопределенного периода времени. HOTP более удобен для пользователя, чем TOTP, поскольку не существует ограничения по времени для ввода OTP.

HOTP требует обновления базы данных каждый раз, когда сервер увеличивает счетчик. Это обновление снижает производительность сервера аутентификации при большой нагрузке. Для повышения эффективности TOTP не запоминает используемые пароли, поэтому нет необходимости выполнять обновления базы данных. Недостатком является возможность повторного использования TOTP в допустимом интервале времени.

Параметры конфигурации TOTP

Алгоритм хеширования OTP
Алгоритм по умолчанию — SHA1. Другие, более безопасные варианты — SHA256 и SHA512.

Количество цифр
Длина OTP. Короткие OTP удобны для пользователя, их легче набирать и легче запоминать. Более длинные OTP более безопасны, чем короткие.

Посмотрите вокруг окна
Количество интервалов, в течение которых сервер пытается сопоставить хэш. Эта опция присутствует в Tuxedo SSO, если часы генератора TOTP или сервера аутентификации выходят из синхронизации. Значение по умолчанию 1 является достаточным. Например, если временной интервал для токена составляет 30 секунд, значение по умолчанию 1 означает, что он будет принимать действительные токены в 90-секундном окне (временной интервал 30 секунд + просмотр вперед 30 секунд + просмотр назад 30 секунд). Каждое увеличение этого значения увеличивает действительное окно на 60 секунд (просмотр вперед 30 секунд + просмотр назад 30 секунд).

Период действия токена OTP
Интервал времени в секундах, в течение которого сервер сопоставляет хэш. Каждый раз, когда интервал проходит, генератор токенов генерирует TOTP.

Многоразовый код
Определите, можно ли повторно использовать токены OTP в процессе аутентификации или пользователю необходимо дождаться следующего токена. Пользователи не могут повторно использовать эти токены по умолчанию, и администратор должен явно указать, что эти токены могут быть повторно использованы.

Параметры конфигурации HOTP

Алгоритм хеширования OTP
Алгоритм по умолчанию — SHA1. Другие, более безопасные варианты — SHA256 и SHA512.

Количество цифр
Длина OTP. Короткие OTP удобны для пользователя, их легче набирать и легче запоминать. Более длинные OTP более безопасны, чем короткие.

Посмотрите вокруг окна
Количество предыдущих и последующих интервалов, которые сервер пытается сопоставить с хешем. Эта опция присутствует в Tuxedo SSO, если часы генератора TOTP или сервера аутентификации рассинхронизируются. Значение по умолчанию 1 является достаточным. Эта опция присутствует в Tuxedo SSO, чтобы покрыть случаи, когда счетчик пользователя опережает сервер.

Начальный счетчик
Значение начального счетчика.

Потоки аутентификации

Поток аутентификации представляет собой контейнер аутентификаций, экранов и действий во время входа в систему, регистрации и других рабочих процессов Tuxedo SSO.

Встроенные потоки

Tuxedo SSO имеет несколько встроенных потоков. Вы не можете изменять эти потоки, но вы можете изменить требования потока в соответствии с вашими потребностями.

Процедура

1. Нажмите «Аутентификация» в меню.

2. Нажмите на элемент «Браузер» в списке, чтобы увидеть подробную информацию.

Поток браузера

Тип аутентификации
Имя аутентификации или действия для выполнения. Если аутентификация имеет отступ, она находится в подпотоке. Она может быть выполнена или нет, в зависимости от поведения ее родителя.

1. Печенье

   При первом успешном входе пользователя Tuxedo SSO устанавливает сеансовый cookie. Если cookie уже установлен, этот тип аутентификации успешен. Поскольку поставщик cookie вернул успех и каждое выполнение на этом уровне потока является альтернативным , Tuxedo SSO не выполняет никаких других выполнений. Это приводит к успешному входу.

2. Керберос

   Этот аутентификатор по умолчанию отключен и пропускается во время работы браузера.

3. Перенаправитель поставщика удостоверений

   Это действие настраивается через ссылку Действия > Конфигурация . Оно перенаправляет на другой IdP для посредничества идентификации .

4. Формы

   Поскольку этот подпоток отмечен как альтернативный , он не будет выполнен, если тип аутентификации Cookie пройден. Этот подпоток содержит дополнительный тип аутентификации, который необходимо выполнить. Tuxedo SSO загружает выполнения для этого подпотока и обрабатывает их.

Первое выполнение — это Форма имени пользователя и пароля , тип аутентификации, который отображает страницу имени пользователя и пароля. Она отмечена как обязательная , поэтому пользователь должен ввести действительное имя пользователя и пароль.

Второе выполнение — это подпоток Browser - Conditional OTP . Этот подпоток является условным и выполняется в зависимости от результата выполнения Condition - User Configured . Если результат истинный, Tuxedo SSO загружает выполнения для этого подпотока и обрабатывает их.

Следующее выполнение — это Condition — User Configured authentication. Эта аутентификация проверяет, настроил ли Tuxedo SSO другие выполнения в потоке для пользователя. Подпоток Browser — Conditional OTP выполняется только тогда, когда у пользователя настроены учетные данные OTP.

Окончательное выполнение — это OTP Form . Tuxedo SSO отмечает это выполнение как обязательное , но оно запускается только тогда, когда у пользователя настроены учетные данные OTP из-за настройки в условном подпотоке. Если нет, пользователь не видит OTP-форму.

Требование
Набор переключателей, управляющих выполнением действия.

Необходимый
Все требуемые элементы в потоке должны быть успешно последовательно выполнены. Поток завершается, если требуемый элемент терпит неудачу.

Альтернатива
Только один элемент должен успешно выполниться, чтобы поток был оценен как успешный. Поскольку Обязательные элементы потока достаточны для того, чтобы отметить поток как успешный, любой Альтернативный элемент потока в потоке, содержащем Обязательные элементы потока, не будет выполнен.

Неполноценный
Элемент не учитывается при отметке потока как успешного.

Условный
Этот тип требования устанавливается только для подпотоков.

- Условный подпоток содержит исполнения. Эти исполнения должны оцениваться логическими утверждениями.

- Если все выполнения оцениваются как истинные , условный подпоток действует как обязательный .
- Если какое-либо выполнение оценивается как false , условный подпоток действует как Disabled .
- Если выполнение не установлено, условный подпоток действует как отключенный .
- Если поток содержит выполнения и для потока не задано значение Conditional , Tuxedo SSO не оценивает выполнения, и они считаются функционально отключенными .

Создание потоков

При проектировании потока следует учитывать важные аспекты функциональности и безопасности.

Чтобы создать поток, выполните следующие действия:

Процедура

1. Нажмите «Аутентификация» в меню.
2. Нажмите Создать поток .

Вы можете скопировать и затем изменить существующий поток. Нажмите «Список действий» (три точки в конце строки), нажмите **«Дублировать** » и введите имя для нового потока.

При создании нового потока необходимо сначала создать поток верхнего уровня со следующими параметрами:

Имя

Название потока.

Описание

Описание, которое вы можете задать для потока.

Тип потока верхнего уровня

Тип потока. Тип client используется только для аутентификации клиентов (приложений). Для всех остальных случаев выбирайте basic .

Создайте поток верхнего уровня

После создания потока Tuxedo SSO отображает кнопки «Добавить шаг» и «Добавить подпоток» .

Пустой новый поток

Поведение потоков и подпотоков определяют три фактора.

- Структура потока и подпотоков.
- Казни в потоках
- Требования, установленные в рамках подпотоков и их исполнения.

Выполнения имеют широкий спектр действий, от отправки письма сброса до проверки OTP. Добавьте выполнения с помощью кнопки Добавить шаг .

Добавление выполнения аутентификации

Выполнения аутентификации могут опционально иметь настроенное опорное значение. Это может использоваться сопоставителем протокола Authentication Method Reference (AMR) для заполнения утверждения amr в токенах доступа и идентификатора OIDC (для получения дополнительной информации о утверждении AMR см. RFC-8176). Когда сопоставитель протокола Authentication Method Reference (AMR) настроен для клиента, он заполнит утверждение amr опорным значением для любого выполнения аутентификатора, которое пользователь успешно завершит во время потока аутентификации.

Добавление ссылочного значения аутентификатора

Существует два типа выполнения: автоматическое выполнение и интерактивное выполнение . Автоматическое выполнение похоже на выполнение Cookie и автоматически выполняет свои действия в потоке. Интерактивное выполнение останавливает поток для получения входных данных. Успешно выполненные выполнения устанавливают свой статус на success . Для завершения потока необходимо как минимум одно выполнение со статусом successful .

Вы можете добавлять подпотоки в потоки верхнего уровня с помощью кнопки Добавить подпоток . Кнопка Добавить подпоток отображает страницу Создать поток выполнения . Эта страница похожа на страницу Создать форму верхнего уровня . Разница в том, что тип потока может быть базовым (по умолчанию) или формой . Тип формы создает подпоток, который генерирует форму для пользователя, похожую на встроенный поток регистрации . Успех подпотоков зависит от того, как оцениваются их выполнения, включая содержащиеся в них подпотоки. Подробное объяснение того, как работают подпотоки, см. в разделе Требования к выполнению .

После добавления выполнения проверьте, что требование имеет правильное значение.

Все элементы в потоке имеют опцию Delete рядом с элементом. Некоторые выполнения имеют пункт меню ⚙ (значок шестеренки) для настройки выполнения. Также можно добавлять выполнения и подпотоки в подпотоки с помощью ссылок Add step и Add sub-flow .

Поскольку порядок выполнения важен, вы можете перемещать выполнения и подпотоки вверх и вниз, перетаскивая их имена.

Обязательно тщательно протестируйте конфигурацию при настройке потока аутентификации, чтобы убедиться в отсутствии дыр в безопасности в вашей настройке. Мы рекомендуем вам протестировать различные угловые случаи. Например, рассмотрите возможность тестирования поведения аутентификации для пользователя, когда вы удаляете различные учетные данные из учетной записи пользователя перед аутентификацией.

Например, когда аутентификаторы 2-го фактора, такие как OTP Form или WebAuthn Authenticator, настроены в потоке как ОБЯЗАТЕЛЬНЫЕ, а у пользователя нет учетных данных определенного типа, пользователь сможет настроить конкретные учетные данные во время самой аутентификации. Такая ситуация означает, что пользователь не аутентифицируется с этими учетными данными, поскольку он настроил их прямо во время аутентификации. Поэтому для аутентификации браузера обязательно настройте свой поток аутентификации с некоторыми учетными данными 1-го фактора, такими как Password или WebAuthn Passwordless Authenticator.

Создание процесса входа в браузер без пароля

Чтобы проиллюстрировать создание потоков, в этом разделе описывается создание расширенного потока входа в браузер. Цель этого потока — предоставить

пользователю выбор между входом в систему без пароля с помощью WebAuthn или двухфакторной аутентификацией с паролем и OTP.

Процедура

1. Нажмите «Аутентификация» в меню.
2. Перейдите на вкладку Потоки .
3. Нажмите Создать поток .
4. Введите Browser Password-lessимя.
5. Нажмите «Создать» .
6. Нажмите Добавить выполнение .
7. Выберите «Cookie» из списка.
8. Нажмите Добавить .
9. Выберите «Альтернативный» для типа аутентификации Cookie , чтобы установить альтернативное требование.
10. Нажмите Добавить шаг .
11. Выберите Kerberos из списка.
12. Нажмите Добавить .
13. Нажмите Добавить шаг .
14. Выберите из списка пункт «Перенаправление поставщика удостоверений» .
15. Нажмите Добавить .
16. Выберите значение «Альтернатива» для типа аутентификации «Перенаправитель поставщика удостоверений», чтобы установить для него альтернативное требование.
17. Нажмите Добавить подпоток .
18. Введите «Формы» в качестве имени.
19. Нажмите Добавить .

20. Выберите «Альтернатива» для типа аутентификации с помощью форм , чтобы установить альтернативное требование.

    Общая часть с потоком браузера

21. Нажмите + меню выполнения форм .
22. Выберите Добавить шаг .
23. Выберите из списка «Форма имени пользователя» .
24. Нажмите Добавить .

На этом этапе форма требует имя пользователя, но не пароль. Мы должны включить аутентификацию по паролю, чтобы избежать рисков безопасности.

1. Нажмите + меню вложенного потока Формы .
2. Нажмите Добавить подпоток .
3. Введите Authenticationимя.
4. Нажмите Добавить .
5. Выберите значение «Обязательно» для типа аутентификации «Аутентификация», чтобы установить ее обязательное требование.
6. Нажмите + меню вложенного потока аутентификации .
7. Нажмите Добавить шаг .
8. Выберите из списка WebAuthn Passwordless Authenticator .
9. Нажмите Добавить .
10. Выберите «Альтернатива» для типа аутентификации Webauthn Passwordless Authenticator, чтобы установить для него альтернативное требование.
11. Нажмите + меню вложенного потока аутентификации .
12. Нажмите Добавить подпоток .
13. Введите Password with OTPимя.
14. Нажмите Добавить .

15. Выберите «Альтернатива» для типа аутентификации «Пароль с OTP», чтобы установить для него альтернативное требование.
16. Нажмите + меню вложенного потока «Пароль с OTP» .
17. Нажмите Добавить шаг .
18. Выберите из списка форму пароля .
19. Нажмите Добавить .
20. Выберите «Обязательно» для типа аутентификации «Форма пароля» , чтобы установить обязательное требование.
21. Нажмите + меню вложенного потока «Пароль с OTP» .
22. Нажмите Добавить шаг .
23. Выберите форму OTP из списка.
24. Нажмите Добавить .
25. Нажмите «Обязательно» для типа аутентификации OTP-формы , чтобы установить его требование как обязательное.

Наконец, поменяйте крепления.

1. Нажмите меню «Действие» в верхней части экрана.
2. Выберите в меню пункт «Привязать поток» .
3. Щелкните раскрывающийся список «Поток браузера» .
4. Нажмите «Сохранить ».

Вход в браузер без пароля

После ввода имени пользователя процесс работает следующим образом:

Если у пользователей записаны учетные данные WebAuthn без пароля, они могут использовать эти учетные данные для прямого входа. Это вход без пароля. Пользователь также может выбрать Password with OTP, поскольку WebAuthn Passwordlessвыполнение и Password with OTPпоток установлены на Alternative .

Если они установлены на Required , пользователь должен ввести WebAuthn, пароль и OTP.

Если пользователь выбирает ссылку Try another way with WebAuthn passwordlessauthentication, пользователь может выбрать между Passwordи Passkey(WebAuthn passwordless). При выборе пароля пользователю необходимо продолжить и войти в систему с назначенным OTP. Если у пользователя нет учетных данных WebAuthn, пользователь должен ввести пароль, а затем OTP. Если у пользователя нет учетных данных OTP, ему будет предложено записать их.

Поскольку выполнение WebAuthn Passwordless установлено на **Alternative** , а не **Required** , этот поток никогда не попросит пользователя зарегистрировать учетные данные WebAuthn. Чтобы у пользователя были учетные данные Webauthn, администратор должен добавить требуемое действие для пользователя. Сделайте это следующим образом:

1. Включение действия **Webauthn Register Passwordless** в области (см. документацию WebAuthn).

2. Настройка необходимого действия с помощью раздела **«Сброс учетных данных» меню управления** учетными данными пользователя .

Создание расширенного потока, такого как этот, может иметь побочные эффекты. Например, если вы включите возможность сброса пароля для пользователей, это будет доступно из формы пароля. В `Reset Credentials`потоке по умолчанию пользователи должны ввести свое имя пользователя. Поскольку пользователь уже ввел имя пользователя ранее в потоке `Browser Password-less`, это действие не нужно для Tuxedo SSO и неоптимально для пользовательского опыта. Чтобы исправить эту проблему, вы можете:

- Дублируйте `Reset Credentials`поток. Задайте ему имя `Reset Credentials for password-less`, например.

- Нажмите **Удалить** (значок корзины) на шаге **Выбор пользователя** .

- В меню **«Действие»** выберите **«Привязать поток»** , затем в раскрывающемся списке выберите **«Сбросить поток учетных данных» и нажмите «Сохранить».**

Создание процесса входа в браузер с пошаговым механизмом

В этом разделе описывается, как создать расширенный поток входа в браузер с использованием механизма step-up. Целью step-up аутентификации является

разрешение доступа к клиентам или ресурсам на основе определенного уровня аутентификации пользователя.

Процедура

1. Нажмите «Аутентификация» в меню.
2. Перейдите на вкладку Потоки .
3. Нажмите Создать поток .
4. Введите Browser Incl Step up Mechanismимя.
5. Нажмите «Сохранить ».
6. Нажмите Добавить выполнение .
7. Выберите «Cookie» из списка.
8. Нажмите Добавить .
9. Выберите «Альтернативный» для типа аутентификации Cookie , чтобы установить альтернативное требование.
10. Нажмите Добавить подпоток .
11. Введите Auth Flow в качестве имени.
12. Нажмите Добавить .
13. Нажмите «Альтернатива» для типа аутентификации Auth Flow , чтобы установить для него альтернативное требование.

Теперь настройте поток для первого уровня аутентификации.

1. Нажмите + меню потока аутентификации .
2. Нажмите Добавить подпоток .
3. Введите 1st Condition Flowимя.
4. Нажмите Добавить .
5. Нажмите Условный для типа аутентификации 1-го потока условий, чтобы установить его требование как условное.

6. Нажмите + меню потока 1-го условия .
7. Нажмите Добавить условие .
8. Выберите из списка Условный — Уровень аутентификации .
9. Нажмите Добавить .
10. Нажмите «Обязательно» для типа аутентификации «Условный — уровень аутентификации», чтобы установить его требование как обязательное.
11. Нажмите ⚙ (значок шестеренки).
12. Введите Level 1псевдоним.
13. Введите 1уровень аутентификации (LoA).
14. Установите Max Age на 36000. Это значение указывается в секундах и эквивалентно 10 часам, что является SSO Session Maxтайм-аутом по умолчанию, установленным в области. В результате, когда пользователь проходит аутентификацию с этим уровнем, последующие входы SSO могут повторно использовать этот уровень, и пользователю не нужно проходить аутентификацию с этим уровнем до конца сеанса пользователя, который по умолчанию составляет 10 часов.
15. Нажмите «Сохранить».

    Настройте условие для первого уровня аутентификации

16. Нажмите + меню потока 1-го условия .
17. Нажмите Добавить шаг .
18. Выберите из списка форму «Имя пользователя и пароль» .
19. Нажмите Добавить .

Теперь настройте поток для второго уровня аутентификации.

1. Нажмите + меню потока аутентификации .
2. Нажмите Добавить подпоток .
3. Введите 2nd Condition Flowпсевдоним.

4. Нажмите Добавить .
5. Нажмите Условный для типа аутентификации 2-го потока условий, чтобы установить его требование как условное.
6. Нажмите + меню 2-го потока условий .
7. Нажмите Добавить условие .
8. Выберите Условный — Уровень аутентификации из списка.
9. Нажмите Добавить .
10. Нажмите «Обязательно» для типа аутентификации «Условный — уровень аутентификации», чтобы установить его требование как обязательное.
11. Нажмите ⚙ (значок шестеренки).
12. Введите Level 2псевдоним.
13. Введите 2уровень аутентификации (LoA).
14. Установите Max Age на 0. В результате, когда пользователь проходит аутентификацию, этот уровень действителен только для текущей аутентификации, но не для последующих аутентификаций SSO. Таким образом, пользователю всегда нужно будет проходить аутентификацию снова с этим уровнем, когда он запрашивается.
15. Нажмите «Сохранить».

    Настройте условие для второго уровня аутентификации

16. Нажмите + меню 2-го потока условий .
17. Нажмите Добавить шаг .
18. Выберите форму OTP из списка.
19. Нажмите Добавить .
20. Нажмите «Обязательно» для типа аутентификации OTP-формы , чтобы установить его требование как обязательное.

Наконец, поменяйте крепления.

1. Нажмите меню «Действие» в верхней части экрана.
2. Выберите из списка пункт «Привязать поток» .
3. В раскрывающемся списке выберите Browser Flow .
4. Нажмите «Сохранить ».

Вход через браузер с механизмом пошагового входа

Запросить определенный уровень аутентификации

Для использования механизма step-up вы указываете требуемый уровень аутентификации (LoA) в своем запросе аутентификации. claimsДля этой цели используется параметр:

https://{DOMAIN}/realms/{REALMNAME}/protocol/openid-connect/auth?client_id={CLIENT-ID}&redirect_uri={REDIRECT-URI}&scope=openid&response_type=code&response_mode=query&nonce=exg16fxdjcu&claims=%7B%22id_token%22%3A%7B%22acr%22%3A%7B%22essential%22%3Atrue%2C%22values%22%3A%5B%22gold%22%5D%7D%7D%7D

Параметр claimsуказывается в представлении JSON:

```
claims= {
            "id_token": {
                "acr": {
                    "essential": true,
                    "values": ["gold"]
                }
            }
        }
```

Адаптер Tuxedo SSO javascript поддерживает простую конструкцию этого JSON и отправку его в запросе на вход. Подробнее см. в разделе Tuxedo SSO JavaScript adapter в разделе о защите приложений .

Вы также можете использовать более простой параметр acr_valuesвместо claimsпараметра, чтобы запросить определенные уровни как несущественные. Это указано в спецификации OIDC.

Вы также можете настроить уровень по умолчанию для конкретного клиента, который используется, когда параметр acr_valuesили
параметр claimsc acrутверждением отсутствует. Для получения дополнительных сведений см. Конфигурация ACR клиента).

> Чтобы запросить acr_values как текст (например, `gold`) вместо числового значения, вы настраиваете сопоставление между ACR и LoA. Его можно настроить на уровне области (рекомендуется) или на уровне клиента. Для настройки см. Сопоставление ACR с LoA .

Более подробную информацию см. в официальной спецификации OIDC .

Логика потока

Логика для предыдущего настроенного потока аутентификации выглядит следующим образом:
если клиент запрашивает высокий уровень аутентификации, то есть Уровень аутентификации 2 (LoA 2), пользователь должен выполнить полную двухфакторную аутентификацию: Имя пользователя/Пароль + OTP. Однако, если у пользователя уже есть сеанс в Tuxedo SSO, в котором он вошел с именем пользователя и паролем (LoA 1), у пользователя запрашивается только второй фактор аутентификации (OTP).

Параметр Max Age в условии определяет, как долго (сколько секунд) действует последующий уровень аутентификации. Этот параметр помогает решить, будет ли пользователю предложено снова представить фактор аутентификации во время последующей аутентификации. Если конкретный уровень X запрашивается параметром claimsили acr_valuesи пользователь уже аутентифицировался с уровнем X, но он истек (например, максимальный возраст настроен на 300, а пользователь аутентифицировался до 310 секунд), то пользователю будет предложено снова пройти повторную аутентификацию с определенным уровнем. Однако если уровень еще не истек, пользователь будет автоматически считаться аутентифицированным с этим уровнем.

Использование Max Age со значением 0 означает, что этот конкретный уровень действителен только для этой единственной аутентификации. Следовательно, каждая повторная аутентификация, запрашивающая этот уровень, должна будет снова проходить аутентификацию с этим уровнем. Это полезно для операций,

требующих более высокой безопасности в приложении (например, отправка платежа) и всегда требующих аутентификации с определенным уровнем.

> Обратите внимание, что такие параметры, как `claims`или `acr_values`могут быть изменены пользователем в URL, когда запрос на вход отправляется клиентом в Tuxedo SSO через браузер пользователя. Эту ситуацию можно смягчить, если клиент использует PAR (Pushed authorization request), объект запроса или другие механизмы, которые не позволяют пользователю перезаписывать параметры в URL. Поэтому после аутентификации клиентам рекомендуется проверять токен ID, чтобы еще раз убедиться, что `acr`токен соответствует ожидаемому уровню.

Если явный уровень не запрошен параметрами, Tuxedo SSO потребует аутентификацию с первым условием LoA, найденным в потоке аутентификации, например, Имя пользователя/Пароль в предыдущем примере. Если пользователь уже был аутентифицирован с этим уровнем и этот уровень истек, пользователю не требуется проходить повторную аутентификацию, но acrв токене будет значение 0. Этот результат считается аутентификацией, основанной исключительно на long-lived browser cookieтом, что указано в разделе 2 спецификации OIDC Core 1.0.

> Во время первой аутентификации пользователя всегда выполняется первый настроенный подпоток с **Conditional - Level Of Authentication** (Условный - Уровень аутентификации), так как у пользователя еще нет никакого уровня. Поэтому мы рекомендуем, чтобы подпоток первого уровня содержал минимально необходимые аутентификаторы для аутентификации пользователя. Кроме того, убедитесь, что подпотоки с различными значениями **Conditional - Level Of Authentication** упорядочены, начиная с самого низкого, как показано в примере выше. Например, если вы настроите подпоток с уровнем 2, а затем добавите еще один подпоток с уровнем 1, подпоток уровня 2 будет всегда запрашиваться во время первой аутентификации, что может быть нежелательным поведением.
>
> Конфликтная ситуация может возникнуть, когда администратор указывает несколько потоков, устанавливает для каждого из них разные уровни LoA и назначает потоки разным клиентам. Однако правило всегда одно и то же: если у пользователя есть определенный уровень, для подключения к клиенту ему достаточно иметь только этот уровень. Администратор должен убедиться, что LoA согласован.

Пример сценария

1. Максимальный возраст установлен на уровне 300 секунд для условия уровня 1.
2. Запрос на вход отправляется без запроса acr. Будет использоваться уровень 1, и пользователю необходимо пройти аутентификацию с именем пользователя и паролем. Токен будет иметь acr=1.

3. Еще один запрос на вход отправляется через 100 секунд. Пользователь автоматически аутентифицируется благодаря SSO, и токен вернется acr=1.
4. Еще один запрос на вход отправляется еще через 201 секунду (301 секунда с момента аутентификации в пункте 2). Пользователь автоматически аутентифицируется благодаря SSO, но токен вернется, acr=0поскольку уровень 1 считается истекшим.
5. Отправлен еще один запрос на вход, но теперь он явно запросит в claimsпараметре ACR уровня 1. Пользователю будет предложено повторно пройти аутентификацию с именем пользователя/паролем, а затем он acr=1будет возвращен в токене.

Требование ACR в токене

Утверждение ACR добавляется к токену с помощью acr loa levelсопоставителя протоколов, определенного в acrобласти клиента. Эта область клиента является областью клиента по умолчанию и, следовательно, будет добавлена ко всем вновь созданным клиентам в области.

Если вам не нужны acrутверждения внутри токенов или вам нужна какая-то специальная логика для их добавления, вы можете удалить область действия клиента из своего клиента.

Обратите внимание, что когда запрос на вход инициирует запрос с claimsпараметром, запрашивающим acrкак essentialутверждение, то Tuxedo SSO всегда будет возвращать один из указанных уровней. Если он не может вернуть один из указанных уровней (например, если запрошенный уровень неизвестен или больше настроенных условий в потоке аутентификации), то Tuxedo SSO выдаст ошибку.

Регистрация или сброс учетных данных, запрошенных клиентом

Обычно, когда пользователь перенаправляется в Tuxedo SSO из клиентского приложения, browserпоток запускается. Этот поток может позволить пользователю зарегистрироваться в случае, если регистрация в области включена, и пользователь нажимает Registerна экран входа. Кроме того, если для области включен параметр «Забыть пароль» , пользователь может нажать Forget passwordна

экран входа, что запускает Reset credentialsпоток, в котором пользователи могут сбросить учетные данные после подтверждения адреса электронной почты.

Иногда может быть полезно, чтобы клиентское приложение напрямую перенаправляло пользователя на экран регистрации или на поток сброса учетных данных . Результирующее действие будет соответствовать действию, когда пользователь нажимает кнопку «Зарегистрироваться» или «Забыть пароль» на обычном экране входа. Автоматическое перенаправление на экран регистрации или сброса учетных данных может быть выполнено следующим образом:

- Когда клиент хочет, чтобы пользователь был перенаправлен непосредственно на регистрацию, клиент OIDC должен заменить самый последний фрагмент из пути URL входа OIDC (/auth) на /registrations. Таким образом, полный URL может быть похож на следующий: https://Tuxedo SSO.example.com/realms/your_realm/protocol/openid-connect/registrations.
- Когда клиент хочет, чтобы пользователь был перенаправлен непосредственно в Reset credentialsпоток, клиент OIDC должен заменить самый последний фрагмент из пути URL-адреса входа OIDC (/auth) на /forgot-credentials.

Предыдущие шаги являются единственным поддерживаемым методом для клиента, чтобы напрямую запросить поток регистрации или сброса учетных данных. В целях безопасности не поддерживается и рекомендуется, чтобы клиентские приложения обходили потоки OIDC/SAML и напрямую перенаправлялись на другие конечные точки Tuxedo SSO (например, конечные точки в `/realms/realm_name/login-actions`или `/realms/realm_name/broker`).

Ограничения сеанса пользователя

Ограничения на количество сеансов, которые может иметь пользователь, можно настроить. Сеансы могут быть ограничены по области или по клиенту.

Чтобы добавить ограничения сеанса в поток, выполните следующие действия.

1. Нажмите «Добавить шаг» для потока.
2. Выберите Ограничитель количества сеансов пользователя из списка элементов.
3. Нажмите Добавить .

4. Нажмите «Обязательно» для типа аутентификации «Ограничитель количества сеансов пользователей», чтобы установить его требование как обязательное.
5. Нажмите ⚙ (значок шестеренки), чтобы открыть ограничитель количества сеансов пользователя .
6. Введите псевдоним для этой конфигурации.
7. Введите требуемое максимальное количество сеансов, которые пользователь может иметь в этой области. Например, если значение равно 2, то 2 сеанса SSO — это максимум, который каждый пользователь может иметь в этой области. Если значение равно 0, эта проверка отключена.
8. Введите требуемое максимальное количество сеансов, которые пользователь может иметь для клиента. Например, если значение равно 2, то 2 сеанса SSO являются максимальным значением в этой области для каждого клиента. Таким образом, когда пользователь пытается пройти аутентификацию на client foo, но этот пользователь уже прошел аутентификацию в 2 сеансах SSO на client foo, либо аутентификация будет отклонена, либо существующий сеанс будет завершен на основе настроенного поведения. Если используется значение 0, эта проверка отключена. Если включены как ограничения сеансов, так и ограничения клиентских сеансов, имеет смысл всегда иметь ограничения клиентских сеансов ниже ограничений сеансов. Лимит на клиента никогда не может превышать лимит всех сеансов SSO этого пользователя.
9. Выберите поведение, которое требуется, когда пользователь пытается создать сеанс после достижения лимита. Доступные поведения:
    - Запретить новый сеанс — при запросе нового сеанса и достижении лимита сеансов новые сеансы создаваться не могут.
    - Завершить самый старый сеанс — при запросе нового сеанса и достижении лимита сеансов самый старый сеанс будет удален и создан новый сеанс.

10. При желании можно добавить пользовательское сообщение об ошибке, которое будет отображаться при достижении лимита.

Обратите внимание, что ограничения сеанса пользователя должны быть добавлены к связанному потоку браузера , потоку прямого предоставления , сбросу учетных данных , а также к любому потоку входа в Post-брокер . Аутентификатор должен быть добавлен в тот момент, когда пользователь уже известен во время аутентификации (обычно в конце потока аутентификации) и обычно должен быть ОБЯЗАТЕЛЬНЫМ. Обратите внимание, что невозможно иметь АЛЬТЕРНАТИВНЫЕ и ОБЯЗАТЕЛЬНЫЕ выполнения на одном уровне.

Для большинства аутентификаторов, таких как Direct grant flow, Reset credentialsили Post broker login flow, рекомендуется добавлять аутентификатор как ОБЯЗАТЕЛЬНЫЙ в конце потока аутентификации. Вот пример потока Reset credentials:

Для Browserпотока рассмотрите возможность не добавлять аутентификатор Session Limits в поток верхнего уровня. Эта рекомендация связана с Cookieаутентификатором, который автоматически повторно аутентифицирует пользователей на основе cookie SSO. Он находится на верхнем уровне, и лучше не проверять ограничения сеанса во время повторной аутентификации SSO, поскольку сеанс пользователя уже существует. Поэтому вместо этого рассмотрите возможность добавления отдельного АЛЬТЕРНАТИВНОГО подпотока, например, следующего authenticate-user-with-session-limitпримера на том же уровне, что и Cookie. Затем вы можете добавить ОБЯЗАТЕЛЬНЫЙ подпоток в следующем real-authentication-subflow`example, as a nested subflow of `authenticate-user-with-session-limitи добавить User Session Limitна том же уровне. Внутри real-authentication-subflowвы можете добавить настоящие аутентификаторы аналогично потоку браузера по умолчанию. Следующий пример потока позволяет пользователям проходить аутентификацию с помощью поставщика удостоверений или с помощью пароля и одноразового пароля:

Что касается Post Broker login flow, вы можете добавить User Session Limitsкак единственный аутентификатор в потоке аутентификации, если у вас нет других аутентификаторов, которые вы запускаете после аутентификации с вашим поставщиком удостоверений. Однако убедитесь, что этот поток настроен как Post

Broker Flowy ваших поставщиков удостоверений. Это требование необходимо, чтобы аутентификация с поставщиками удостоверений также участвовала в ограничениях сеанса.

> В настоящее время администратор несет ответственность за поддержание согласованности между различными конфигурациями. Поэтому убедитесь, что все ваши потоки используют одну и ту же конфигурацию `User Session Limits`.
>
> Функция ограничения сеанса пользователя недоступна для CIBA.

Скрипт Аутентификатор

Возможность загрузки скриптов через консоль администратора и конечные точки REST устарела.

Более подробную информацию см. в разделе Поставщики JavaScript .

Керберос

Tuxedo SSO поддерживает вход с билетом Kerberos через протокол Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO). SPNEGO прозрачно аутентифицируется через веб-браузер после того, как пользователь аутентифицирует сеанс. Для случаев, не связанных с веб-доступом, или когда билет недоступен во время входа, Tuxedo SSO поддерживает вход с именем пользователя и паролем Kerberos.

Типичный вариант использования веб-аутентификации:

1. Пользователь входит в систему на рабочем столе.
2. Пользователь получает доступ к веб-приложению, защищенному Tuxedo SSO, с помощью браузера.
3. Приложение перенаправляет на страницу входа в Tuxedo SSO.
4. Tuxedo SSO отображает HTML-экран входа со статусом 401 и заголовком HTTPWWW-Authenticate: Negotiate
5. Если браузер имеет билет Kerberos от входа на рабочем столе, браузер передает информацию о входе на рабочем столе в Tuxedo SSO в заголовке Authorization: Negotiate 'spnego-token'. В противном случае он

отображает стандартный экран входа, и пользователь вводит учетные данные для входа.

6. Tuxedo SSO проверяет токен из браузера и аутентифицирует пользователя.
7. Если используется LDAPFederationProvider с поддержкой аутентификации Kerberos, Tuxedo SSO предоставляет данные пользователя из LDAP. Если используется KerberosFederationProvider, Tuxedo SSO позволяет пользователю обновлять профиль и предварительно заполнять данные для входа.
8. Tuxedo SSO возвращается в приложение. Tuxedo SSO и приложение взаимодействуют через OpenID Connect или сообщения SAML. Tuxedo SSO действует как брокер для входа Kerberos/SPNEGO. Поэтому аутентификация Tuxedo SSO через Kerberos скрыта от приложения.

Схема Negotiate www-authenticate допускает NTLM в качестве резервного варианта Kerberos, а в некоторых веб-браузерах в Windows NTLM поддерживается по умолчанию. Если вызов www-authenticate поступает с сервера, не входящего в список разрешенных браузеров, пользователи могут столкнуться с диалоговым окном NTLM. Пользователю необходимо нажать кнопку отмены в диалоговом окне, чтобы продолжить, поскольку Tuxedo SSO не поддерживает этот механизм. Такая ситуация может возникнуть, если веб-браузеры интрасети не настроены строго или если Tuxedo SSO обслуживает пользователей как в интрасети, так и в Интернете. Пользовательский аутентификатор может использоваться для ограничения вызовов Negotiate белым списком хостов.

Для настройки аутентификации Kerberos выполните следующие действия:

1. Настройка и конфигурирование сервера Kerberos (KDC).
2. Настройка и конфигурирование сервера Tuxedo SSO.
3. Настройка и конфигурирование клиентских машин.

Настройка сервера Kerberos

Шаги по настройке сервера Kerberos зависят от операционной системы (ОС) и поставщика Kerberos. Ознакомьтесь с документацией по Windows Active Directory, MIT Kerberos и вашей ОС для получения инструкций по настройке и конфигурированию сервера Kerberos.

Во время настройки выполните следующие действия:

1. Добавьте несколько пользовательских принципов в вашу базу данных Kerberos. Вы также можете интегрировать свой Kerberos с LDAP, чтобы учетные записи пользователей предоставлялись с сервера LDAP.
2. Добавьте принципала службы для службы "HTTP". Например, если сервер Tuxedo SSO работает на www.mydomain.org, добавьте принципала службы HTTP/www.mydomain.org@<kerberos realm>.

   На MIT Kerberos вы запускаете сеанс "kadmin". На машине с MIT Kerberos вы можете использовать команду:

sudo kadmin.local

Затем добавьте HTTP-принципал и экспортируйте его ключ в файл keytab с помощью таких команд:

addprinc -randkey HTTP/www.mydomain.org@MYDOMAIN.ORG
ktadd -k /tmp/http.keytab HTTP/www.mydomain.org@MYDOMAIN.ORG

Убедитесь, что файл keytab /tmp/http.keytabдоступен на хосте, где запущен Tuxedo SSO.

Установка и настройка сервера Tuxedo SSO

Установите клиент Kerberos на свой компьютер.

Процедура

1. Установите клиент Kerberos. Если на вашем компьютере установлена Fedora, Ubuntu или RHEL, установите пакет freeipa-client , содержащий клиент Kerberos и другие утилиты.
2. Настройте клиент Kerberos (в Linux параметры конфигурации находятся в файле /etc/krb5.conf).

   Добавьте свою область Kerberos в конфигурацию и настройте HTTP-домены, на которых работает ваш сервер.

   Например, для области MYDOMAIN.ORG вы можете настроить domain_realmраздел следующим образом:

   [domain_realm]

```
.mydomain.org = MYDOMAIN.ORG
mydomain.org = MYDOMAIN.ORG
```

3. Экспортируйте файл keytab с HTTP-принципалом и убедитесь, что файл доступен процессу, запускающему сервер Tuxedo SSO. Для производства убедитесь, что файл доступен для чтения только этим процессом.

   Для примера MIT Kerberos выше мы экспортировали keytab в /tmp/http.keytabфайл. Если ваш Key Distribution Centre (KDC) и Tuxedo SSO работают на одном хосте, файл уже доступен.

Включение обработки SPNEGO
По умолчанию Tuxedo SSO отключает поддержку протокола SPNEGO. Чтобы включить его, перейдите в поток браузера и включите Kerberos .

Поток браузера

Установите требование Kerberos с отключенного на альтернативное (Kerberos необязателен) или обязательное (браузер должен иметь включенный Kerberos). Если вы не настроили браузер для работы с SPNEGO или Kerberos, Tuxedo SSO вернется к обычному экрану входа.

Настройка поставщиков федерации хранилищ пользователей Kerberos
Теперь вам необходимо использовать User Storage Federation для настройки того, как Tuxedo SSO интерпретирует билеты Kerberos. Существуют два разных поставщика федерации с поддержкой аутентификации Kerberos.

Для аутентификации с помощью Kerberos, поддерживаемого сервером LDAP, настройте поставщика федерации LDAP .

Процедура

1. Перейдите на страницу конфигурации вашего провайдера LDAP.

   Интеграция Ldap Kerberos

2. Переключите Разрешить аутентификацию Kerberos в положение ВКЛ.

Если разрешить аутентификацию Kerberos, Tuxedo SSO будет использовать информацию о пользователе основного доступа Kerberos, чтобы информацию можно было импортировать в среду Tuxedo SSO.

Если сервер LDAP не выполняет резервное копирование вашего решения Kerberos, используйте поставщика федерации хранения данных пользователей Kerberos .

Процедура

1. Нажмите «Федерация пользователей» в меню.
2. Выберите Kerberos в поле «Добавить поставщика» .

Поставщик хранилища пользователей Kerberos

Поставщик Kerberos анализирует билет Kerberos для простой информации о принципале и импортирует информацию в локальную базу данных Tuxedo SSO. Информация о профиле пользователя, такая как имя, фамилия и адрес электронной почты, не предоставляется.

Настройка и конфигурирование клиентских машин

Клиентские машины должны иметь клиент Kerberos и настроить его, krb5.confкак описано выше . Клиентские машины также должны включить поддержку входа SPNEGO в своем браузере. См. настройку Firefox для Kerberos, если вы используете браузер Firefox.

URI .mydomain.orgдолжен быть указан в network.negotiate-auth.trusted-urisпараметрах конфигурации.

В доменах Windows клиентам не нужно настраивать свою конфигурацию. Internet Explorer и Edge уже могут участвовать в аутентификации SPNEGO.

Примеры установок

Образ докера Tuxedo SSO и FreeIPA

При установке docker запустите образ docker с установленным сервером FreeIPA. FreeIPA предоставляет интегрированное решение безопасности с MIT Kerberos и сервером LDAP 389. Образ также содержит сервер Tuxedo SSO, настроенный с

поставщиком LDAP Federation и включенной аутентификацией SPNEGO/Kerberos на сервере FreeIPA. Подробности см. здесь .

ApacheDS тестирует сервер Kerberos
Для быстрого тестирования и модульных тестов используйте простой сервер ApacheDS Kerberos. Вам необходимо собрать Tuxedo SSO из исходного кода, а затем запустить сервер Kerberos с maven-exec-plugin из нашего тестового набора. Подробности см. здесь .

Делегирование полномочий

Kerberos поддерживает делегирование учетных данных. Приложениям может потребоваться доступ к билету Kerberos, чтобы они могли повторно использовать его для взаимодействия с другими службами, защищенными Kerberos. Поскольку сервер Tuxedo SSO обработал протокол SPNEGO, вы должны распространить учетные данные GSS в своем приложении в утверждении токена OpenID Connect или атрибуте утверждения SAML. Tuxedo SSO передает это в ваше приложение с сервера Tuxedo SSO. Чтобы вставить это утверждение в токен или утверждение, каждое приложение должно включить встроенный сопоставитель протоколов gss delegation credential. Этот сопоставитель доступен на вкладке Сопоставители на странице клиента приложения. Подробнее см. в главе Сопоставители протоколов .

Приложения должны десериализовать утверждение, которое они получают от Tuxedo SSO, прежде чем использовать его для вызовов GSS против других служб. Когда вы десериализуете учетные данные из токена доступа в объект GSSCredential, создайте GSSContext с этими учетными данными, переданными в GSSManager.createContextметод. Например:

```
// Obtain accessToken in your application.
Tuxedo SSOPrincipal Tuxedo SSOPrincipal = (Tuxedo SSOPrincipal)
servletReq.getUserPrincipal();
AccessToken accessToken = Tuxedo SSOPrincipal.getTuxedo
SSOSecurityContext().getToken();

// Retrieve Kerberos credential from accessToken and deserialize it
String serializedGssCredential = (String) accessToken.getOtherClaims().
```

```
    get(org.Tuxedo
SSO.common.constants.KerberosConstants.GSS_DELEGATION_CREDENTIAL);

GSSCredential deserializedGssCredential = org.Tuxedo
SSO.common.util.KerberosSerializationUtils.
    deserializeCredential(serializedGssCredential);

// Create GSSContext to call other Kerberos-secured services
GSSContext context = gssManager.createContext(serviceName, krb5Oid,
    deserializedGssCredential, GSSContext.DEFAULT_LIFETIME);
```

Настройте `forwardable`билеты Kerberos в `krb5.conf`файле и добавьте поддержку делегированных учетных данных в ваш браузер.

Делегирование учетных данных имеет последствия для безопасности, поэтому используйте его только при необходимости и только с HTTPS. Подробнее и с примером смотрите эту статью .

Межрегиональное доверие

В протоколе Kerberos realmесть набор принципалов Kerberos. Определение этих принципалов существует в базе данных Kerberos, которая обычно является сервером LDAP.

Протокол Kerberos допускает доверие между областями. Например, если существуют 2 области Kerberos, A и B, то доверие между областями позволит пользователям из области A получать доступ к ресурсам области B. Область B доверяет области A.

Межобластное доверие Kerberos

Сервер Tuxedo SSO поддерживает кросс-реалмное доверие. Для реализации этого выполните следующее:

- Настройте серверы Kerberos для межобластного доверия. Реализация этого шага зависит от реализаций сервера Kerberos. Этот шаг необходим для добавления принципала Kerberos krbtgt/B@Aв базы данных Kerberos областей A и B. Этот принципал должен иметь одинаковые ключи в обеих областях Kerberos. Принципалы должны иметь одинаковые пароли, номера

версий ключей и шифры в обеих областях. Более подробную информацию см. в документации сервера Kerberos.

По умолчанию доверие между областями является однонаправленным. Необходимо добавить принципала `krbtgt/A@B`в обе базы данных Kerberos для двунаправленного доверия между областью A и областью B. Однако доверие по умолчанию является транзитивным. Если область B доверяет области A, а область C доверяет области B, то область C доверяет области A без принципала, `krbtgt/C@A`, доступного. На стороне клиента Kerberos может потребоваться дополнительная настройка (например, `capaths`), чтобы клиенты могли найти путь доверия. Более подробную информацию см. в документации Kerberos.

- Настроить сервер Tuxedo SSO
    - При использовании поставщика хранилища LDAP с поддержкой Kerberos настройте принципала сервера для области B, как в этом примере: HTTP/mydomain.com@B. Сервер LDAP должен найти пользователей из области A, если пользователи из области A должны успешно пройти аутентификацию в Tuxedo SSO, поскольку Tuxedo SSO должен выполнить поток SPNEGO, а затем найти пользователей.

Поиск пользователей основан на параметре поставщика хранилища LDAP Kerberos principal attribute. Если он настроен, например, со значением вроде userPrincipalName, то после аутентификации SPNEGO пользователя john@ATuxedo SSO попытается найти пользователя LDAP с атрибутом, userPrincipalNameэквивалентным john@A. Если Kerberos principal attributeоставить пустым, то Tuxedo SSO будет искать пользователя LDAP на основе префикса его принципала kerberos с пропущенной областью. Например, основной пользователь Kerberos john@Aдолжен быть доступен в LDAP под именем пользователя john, поэтому обычно под LDAP DN, таким как uid=john,ou=People,dc=example,dc=com. Если вы хотите, чтобы пользователи из областей A и B проходили аутентификацию, убедитесь, что LDAP может найти пользователей из обеих областей A и B.

- При использовании поставщика хранилища пользователей Kerberos (обычно Kerberos без интеграции LDAP) настройте принципала сервера как HTTP/mydomain.com@B, и пользователи из областей Kerberos A и B должны иметь возможность проходить аутентификацию.

Пользователи из нескольких областей Kerberos могут проходить аутентификацию, поскольку у каждого пользователя будет
атрибут, KERBEROS_PRINCIPALссылающийся на принципал Kerberos, используемый для аутентификации, и это используется для дальнейшего поиска этого пользователя. Чтобы избежать конфликтов, когда пользователь находится johnкак в областях Kerberos A, так и B, имя пользователя Tuxedo SSO может содержать область Kerberos в нижнем регистре. Например, имя пользователя будет john@a. На всякий случай, когда область совпадает с настроенным Kerberos realm, суффикс области может быть опущен из сгенерированного имени пользователя. Например, имя пользователя будет johnдля принципала Kerberos john@A, если Kerberos realmнастроено на поставщике Kerberos A.

Поиск неисправностей

Если у вас возникли проблемы, включите дополнительное ведение журнала для устранения неполадки:

- Включите Debugфлаг в консоли администратора для поставщиков федерации Kerberos или LDAP
- Включите ведение журнала TRACE для категории org.Tuxedo SSO, чтобы получать больше информации в журналах сервера
- Добавьте системные свойства -Dsun.security.krb5.debug=trueи-Dsun.security.spnego.debug=true

Аутентификация пользователя клиентского сертификата X.509

Tuxedo SSO поддерживает вход с использованием клиентского сертификата X.509, если вы настроили сервер на использование взаимной аутентификации SSL.

Типичный рабочий процесс:

- Клиент отправляет запрос аутентификации по каналу SSL/TLS.
- Во время установления связи SSL/TLS сервер и клиент обмениваются сертификатами x.509/v3.

- Контейнер (WildFly) проверяет путь PKIX сертификата и дату истечения срока действия сертификата.
- Аутентификатор клиентского сертификата x.509 проверяет клиентский сертификат, используя следующие методы:
    - Проверяет статус отзыва сертификата с помощью CRL или точек распространения CRL.
    - Проверяет статус отзыва сертификата с помощью OCSP (Online Certificate Status Protocol).
    - Проверяет, соответствует ли ключ в сертификате ожидаемому ключу.
    - Проверяет, соответствует ли расширенный ключ в сертификате ожидаемому расширенному ключу.
- Если любая из этих проверок не пройдена, аутентификация x.509 не пройдена. В противном случае аутентификатор извлекает идентификатор сертификата и сопоставляет его с существующим пользователем.

Когда сертификат сопоставляется с существующим пользователем, поведение различается в зависимости от потока аутентификации:

- В браузерном потоке сервер предлагает пользователям подтвердить свою личность или войти в систему, указав имя пользователя и пароль.
- В потоке прямого предоставления сервер регистрирует пользователя.

Обратите внимание, что проверка пути сертификата PKIX является обязанностью веб-контейнера. Аутентификатор X.509 на стороне Tuxedo SSO обеспечивает только дополнительную поддержку для проверки срока действия сертификата, статуса отзыва сертификата и использования ключа. Если вы используете Tuxedo SSO, развернутый за обратным прокси-сервером, убедитесь, что ваш обратный прокси-сервер настроен на проверку пути PKIX. Если вы не используете обратный прокси-сервер и пользователи напрямую обращаются к WildFly, все должно быть в порядке, поскольку WildFly обеспечивает проверку пути PKIX, если он настроен так, как описано ниже.

Функции

Поддерживаемые источники удостоверений сертификатов:

- Сопоставьте SubjectDN с помощью регулярных выражений

- Атрибут адреса электронной почты субъекта X500
- X500 Адрес электронной почты субъекта из расширения альтернативного имени субъекта (RFC822Name General Name)
- X500 Другое имя субъекта из Subject Alternative Name Extension. Это другое имя — User Principal Name (UPN), как правило.
- Атрибут общего имени субъекта X500
- Сопоставьте IssuerDN с помощью регулярных выражений
- Серийный номер сертификата
- Серийный номер сертификата и IssuerDN
- Отпечаток сертификата SHA-256
- Полный сертификат в формате PEM

Регулярные выражения
Tuxedo SSO извлекает идентификатор сертификата из Subject DN или Issuer DN, используя регулярное выражение в качестве фильтра. Например, это регулярное выражение соответствует атрибуту email:

emailAddress=(.*?)(?:,|$)

Фильтрация по регулярному выражению применяется, если Identity Sourcезадано значение Match SubjectDN using regular expressionили Match IssuerDN using regular expression.

Сопоставление идентификатора сертификата с существующим пользователем
Сопоставление идентификатора сертификата может сопоставлять извлеченный идентификатор пользователя с существующим именем пользователя, адресом электронной почты или пользовательским атрибутом, значение которого соответствует идентификатору сертификата. Например, установка Identity sourcena адрес электронной почты субъекта или User mapping methodна имя пользователя или адрес электронной почты заставляет аутентификатор клиентского сертификата X.509 использовать атрибут электронной почты в DN

субъекта сертификата в качестве критерия поиска при поиске существующего пользователя по имени пользователя или адресу электронной почты.

- Если вы отключите **Login with email** в настройках области, те же правила применяются к аутентификации сертификата. Пользователи не смогут войти, используя атрибут email.
- Для использования `Certificate Serial Number and IssuerDN`в качестве источника идентификации требуются два настраиваемых атрибута: серийный номер и IssuerDN.
- `SHA-256 Certificate thumbprint`— это строчное шестнадцатеричное представление отпечатка сертификата SHA-256.
- Использование `Full certificate in PEM format`в качестве источника идентификации ограничено пользовательскими атрибутами, сопоставленными с внешними источниками федерации, такими как LDAP. Tuxedo SSO не может хранить сертификаты в своей базе данных из-за ограничений по длине, поэтому в случае LDAP необходимо включить `Always Read Value From LDAP`.

Расширенная проверка сертификата

- Проверка статуса отзыва с использованием CRL.
- Проверка статуса отзыва с использованием CRL/Distribution Point.
- Проверка статуса отзыва с использованием OCSP/Responder URI.
- Проверка использования ключа сертификата.
- Проверка ExtendedKeyUsage сертификата.

Добавление аутентификации клиентского сертификата X.509 в потоки браузера

1. Нажмите «Аутентификация» в меню.
2. Нажмите на ссылку «Браузер» .
3. В списке действий выберите Дублировать .
4. Введите имя копии.
5. Нажмите «Дублировать» .
6. Нажмите Добавить шаг .
7. Нажмите «X509/Форма проверки имени пользователя».

8. Нажмите Добавить .

   Исполнение X509

9. Щелкните и перетащите «Форму X509/Проверка имени пользователя» поверх выполнения «Формы браузера».
10. Установите требование «АЛЬТЕРНАТИВА».

    Поток браузера X509

11. Нажмите меню «Действие» .
12. Нажмите кнопку «Привязать поток» .
13. В раскрывающемся списке выберите пункт «Браузер» .
14. Нажмите «Сохранить ».

    Привязки потока браузера X509

Настройка аутентификации клиентского сертификата X.509

Конфигурация X509

Источник идентификации пользователя

> Определяет метод извлечения идентификатора пользователя из клиентского сертификата.

Включено каноническое представление DN

> Определяет, использовать ли канонический формат для определения отличительного имени. Официальная документация Java API описывает формат. Этот параметр влияет на два источника идентификации пользователя: Match SubjectDN using regular expression и Match IssuerDN using regular expression only. Включите этот параметр при настройке нового экземпляра Tuxedo SSO. Отключите этот параметр, чтобы сохранить обратную совместимость с существующими экземплярами Tuxedo SSO.

Включить шестнадцатеричное представление серийного номера

Представьте серийный номер в шестнадцатеричном формате. Серийный номер со знаковым битом, установленным на 1, должен быть дополнен слева октетом 00. Например, серийный номер с десятичным значением 161 или a1 в шестнадцатеричном формате кодируется как 00a1 в соответствии с RFC5280. Подробнее см. RFC5280, приложение B.

Регулярное выражение

Регулярное выражение для использования в качестве фильтра для извлечения идентификатора сертификата. Выражение должно содержать одну группу.

Метод сопоставления пользователей

Определяет метод сопоставления удостоверения сертификата с существующим пользователем. Имя пользователя или адрес электронной почты выполняет поиск существующих пользователей по имени пользователя или адресу электронной почты. Пользовательский атрибутный сопоставитель выполняет поиск существующих пользователей с пользовательским атрибутом, соответствующим удостоверению сертификата. Имя пользовательского атрибута можно настроить.

Имя атрибута пользователя

Пользовательский атрибут, значение которого соответствует идентификатору сертификата. Используйте несколько пользовательских атрибутов, когда сопоставление атрибутов связано с несколькими значениями, например, «Серийный номер сертификата и IssuerDN».

Проверка CRL включена

Проверьте статус отзыва сертификата с помощью списка отзыва сертификатов. Местоположение списка определяется в атрибуте пути к файлу CRL .

Включить точку распространения CRL для проверки статуса отзыва сертификата

Используйте CDP для проверки статуса отзыва сертификата. Большинство органов PKI включают CDP в свои сертификаты.

Путь к файлу CRL

Путь к файлу, содержащему список CRL. Значение должно быть путем к допустимому файлу, если включена опция CRL Checking Enabled .

Проверка OCSP включена

Проверяет статус отзыва сертификата с помощью протокола онлайн-статуса сертификата.

Поведение OCSP при отказе

По умолчанию проверка OCSP должна возвращать положительный ответ для продолжения успешной аутентификации. Однако иногда эта проверка может быть неубедительной: например, сервер OCSP может быть недоступен, перегружен или сертификат клиента может не содержать URI ответчика OCSP. Когда этот параметр включен, аутентификация будет отклонена только в том случае, если ответчик OCSP получит явный отрицательный ответ и сертификат будет определенно отозван. Если допустимый ответ OCSP недоступен, попытка аутентификации будет принята.

URI ответчика OCSP

Переопределить значение URI ответчика OCSP в сертификате.

Проверить использование ключа

Проверяет, установлены ли биты расширения KeyUsage сертификата. Например, "digitalSignature,KeyEncipherment" проверяет, установлены ли биты 0 и 2 в расширении KeyUsage. Оставьте этот параметр пустым, чтобы отключить проверку Key Usage. См. RFC5280, Section-4.2.1.3 для получения дополнительной информации. Tuxedo SSO выдает ошибку, если происходит несоответствие использования ключа.

Проверить использование расширенного ключа

Проверяет одну или несколько целей, определенных в расширении Extended Key Usage. См. RFC5280, Section-4.2.1.12 для получения дополнительной информации. Оставьте этот параметр пустым, чтобы отключить проверку Extended Key Usage. Tuxedo SSO выдает ошибку, когда помечается как

критический выдающим CA и происходит несоответствие расширения использования ключа.

Проверить политику сертификата

Проверяет один или несколько идентификаторов OID политики, как определено в расширении политики сертификата. См. RFC5280, раздел-4.2.1.4 . Оставьте параметр пустым, чтобы отключить проверку политики сертификата. Несколько политик следует разделять запятой.

Режим проверки политики сертификата

Если в настройке указано более одной политики Validate Certificate Policy, она решает, должно ли сопоставление проверять наличие всех запрошенных политик или для успешной аутентификации достаточно одного сопоставления. Значение по умолчанию — All, что означает, что все запрошенные политики должны присутствовать в клиентском сертификате.

Обход подтверждения личности

Если включено, аутентификация сертификата клиента X.509 не запрашивает у пользователя подтверждение идентичности сертификата. Tuxedo SSO подписывает пользователя после успешной аутентификации.

Повторная проверка сертификата клиента

Если установлено, цепочка доверия клиентских сертификатов всегда будет проверяться на уровне приложения с использованием сертификатов, имеющихся в настроенном хранилище доверия. Это может быть полезно, если базовый веб-сервер не обеспечивает проверку цепочки клиентских сертификатов, например, потому что он находится за невалидирующим балансировщиком нагрузки или обратным прокси-сервером, или когда количество разрешенных CA слишком велико для взаимного согласования SSL (большинство браузеров ограничивают максимальный размер пакета согласования SSL 32767 байтами, что соответствует примерно 200 объявленным CA). По умолчанию эта опция отключена.

Добавление аутентификации клиентского сертификата X.509 в поток прямого предоставления

1. Нажмите «Аутентификация» в меню.
2. Выберите «Дублировать» в «Списке действий», чтобы создать копию встроенного потока «Прямое предоставление».
3. Введите имя копии.
4. Нажмите «Дублировать» .
5. Щелкните созданный поток.
6. Нажмите на значок корзины 🗑 в разделе «Проверка имени пользователя» и нажмите «Удалить » .
7. Нажмите на значок корзины 🗑 рядом с «Паролем» и нажмите «Удалить » .
8. Нажмите Добавить шаг .
9. Нажмите «X509/Проверить имя пользователя».
10. Нажмите Добавить .

    X509 прямое исполнение гранта

11. Настройте конфигурацию аутентификации x509, выполнив шаги, описанные в разделе «Последовательность действий браузера x509» .
12. Перейдите на вкладку Привязки .
13. Нажмите на раскрывающийся список «Прямой поток грантов» .
14. Нажмите на недавно созданный поток «x509 Direct Grant».
15. Нажмите «Сохранить ».

    X509 прямые привязки потока грантов

Пример использования CURL
Следующий пример показывает, как получить токен доступа для пользователя в области testc прямым потоком предоставления. Пример использует OAuth2

Resource Owner Password Credentials Grant в разделе защищенных приложений и конфиденциального клиента resource-owner:

```
curl \
  -d "client_id=resource-owner" \
  -d "client_secret=40cc097b-2a57-4c17-b36a-8fdf3fc2d578" \
  -d "grant_type=password" \
  --cacert /tmp/truststore.pem \
  --cert /tmp/keystore.pem:kssecret \
  "https://localhost:8543/realms/test/protocol/openid-connect/token"
```

Файл /tmp/truststore.pemуказывает на файл с truststore, содержащим сертификат сервера Tuxedo SSO. Файл /tmp/keystore.pemсодержит закрытый ключ и сертификаты, соответствующие пользователю Tuxedo SSO, которые будут успешно аутентифицированы этим запросом. От конфигурации аутентификатора зависит, как именно содержимое сертификата сопоставляется с пользователем Tuxedo SSO, как описано в разделе конфигурации . Это kssecretможет быть пароль этого файла keystore.

В зависимости от вашей среды может потребоваться использовать больше параметров для команд CURL, например:

- Вариант --insecure, если вы используете самоподписанные сертификаты
- Возможность --capathвключить весь каталог, содержащий путь к центру сертификации
- Параметры --cert-typeили --key-typeесли вы хотите использовать файлы, отличные отPEM

Пожалуйста, обратитесь к документации инструмента curlдля получения подробной информации, если это необходимо. Если вы используете другие инструменты, кроме curl, обратитесь к документации вашего инструмента. Однако настройка будет аналогичной. Существует необходимость включить хранилище ключей и доверенное хранилище, а также учетные данные клиента, если вы используете конфиденциальный клиент.

> Если это возможно, предпочтительнее использовать учетные записи служб вместе с аутентификацией клиента MTLS (аутентификатор клиента `X509 Certificate`), а не

использовать прямой грант с аутентификацией X.509, поскольку прямой грант может потребовать совместного использования сертификата пользователя с клиентскими приложениями. При использовании учетной записи службы токены получаются от имени самого клиента, что в целом является лучшей и более безопасной практикой.

Веб-аутентификация W3C (WebAuthn)

Tuxedo SSO обеспечивает поддержку W3C Web Authentication (WebAuthn) . Tuxedo SSO работает как проверяющая сторона (RP) WebAuthn .

Успех операций WebAuthn зависит от аутентификатора, браузера и платформы, поддерживающих WebAuthn пользователя. Убедитесь, что ваш аутентификатор, браузер и платформа поддерживают спецификацию WebAuthn.

Настраивать

Процедура настройки поддержки WebAuthn для 2FA следующая:

Включить регистрацию аутентификатора WebAuthn

1. Нажмите «Аутентификация» в меню.
2. Перейдите на вкладку «Требуемые действия» .
3. Переведите переключатель Webauthn Register в положение ON .

Установите переключатель «Действие по умолчанию» в положение «ВКЛ», если вы хотите, чтобы все новые пользователи регистрировали свои учетные данные WebAuthn.

Добавление аутентификации WebAuthn в поток браузера

1. Нажмите «Аутентификация» в меню.
2. Нажмите на ссылку «Браузер» .
3. Выберите «Дублировать» в «Списке действий», чтобы создать копию встроенного потока браузера .
4. Введите «WebAuthn Browser» в качестве имени копии.
5. Нажмите «Дублировать» .
6. Нажмите на имя, чтобы перейти к подробностям

7. Нажмите на значок корзины 🗑 в разделе «Браузер WebAuthn — условный одноразовый пароль» и нажмите «Удалить » .

Если вам требуется WebAuthn для всех пользователей:

1. Нажмите + меню форм браузера WebAuthn .
2. Нажмите Добавить шаг .
3. Нажмите Аутентификатор WebAuthn .
4. Нажмите Добавить .
5. Выберите значение «Обязательно» для типа аутентификации WebAuthn Authenticator , чтобы установить его требование как обязательное.
6. Нажмите меню «Действие» в верхней части экрана.
7. В раскрывающемся списке выберите пункт Привязать поток .
8. Выберите Браузер из раскрывающегося списка.
9. Нажмите «Сохранить ».

Если у пользователя нет учетных данных WebAuthn, он должен зарегистрировать учетные данные WebAuthn.

Пользователи могут войти с помощью WebAuthn, если у них есть зарегистрированные учетные данные WebAuthn. Поэтому вместо добавления выполнения WebAuthn Authenticator вы можете:

Процедура

1. Нажмите + меню в строке WebAuthn Browser Forms .
2. Нажмите Добавить подпоток .
3. В поле имени введите «Условная 2FA» .
4. Выберите Условный для Условной 2FA , чтобы сделать ее требование условным.
5. В строке «Условная 2FA» нажмите на знак «плюс» + и выберите «Добавить условие» .

6. Нажмите Добавить условие .
7. Выберите Условие — Настраивается пользователем .
8. Нажмите Добавить .
9. Выберите «Обязательно» для параметра «Условие — Настраивается пользователем», чтобы установить его требование как обязательное.
10. Перетащите аутентификатор WebAuthn в поток условной 2FA.
11. Выберите «Альтернатива» для аутентификатора WebAuthn, чтобы установить для него альтернативное требование.

Пользователь может выбрать между использованием WebAuthn и OTP для второго фактора:

Процедура

1. В строке «Условная 2FA» нажмите на знак «плюс» + и выберите «Добавить шаг» .
2. Выберите форму OTP из списка.
3. Нажмите Добавить .
4. Выберите «Альтернатива» для формы одноразового пароля , чтобы установить для нее альтернативное требование.

Аутентификация с помощью аутентификатора WebAuthn

После регистрации аутентификатора WebAuthn пользователь выполняет следующие операции:

- Откройте форму входа. Пользователь должен пройти аутентификацию с помощью имени пользователя и пароля.
- Браузер пользователя просит его пройти аутентификацию с помощью аутентификатора WebAuthn.

Управление WebAuthn в качестве администратора

Управление учетными данными

Tuxedo SSO управляет учетными данными WebAuthn аналогично другим учетным данным из раздела «Управление учетными данными пользователей» :

- Tuxedo SSO назначает пользователям необходимое действие для создания учетных данных WebAuthn из списка «Действия по сбросу» и выбирает «Регистрация WebAuthn» .
- Администраторы могут удалить учетные данные WebAuthn, нажав «Удалить» .
- Администраторы могут просматривать данные учетных данных, такие как AAGUID, выбрав Показать данные… .
- Администраторы могут задать метку для учетных данных, указав значение в поле «Метка пользователя» и сохранив данные.

Управление политикой

Администраторы могут настраивать операции, связанные с WebAuthn, как политику WebAuthn для каждой области.

Процедура

1. Нажмите «Аутентификация». в меню.
2. Нажмите на политику .
3. Нажмите «Политика WebAuthn». .
4. Настройте элементы политики (см. описание ниже).
5. Нажмите «Сохранить ».

Ниже приведены настраиваемые элементы и их описание:

Конфигурация	Описание
Название проверяющей стороны	Читаемое имя сервера как проверяющей стороны WebAuthn. Этот элемент является обязательным и применяется к регистрации аутентификатора WebAuthn. Значение по умолчанию — «Tuxedo SSO». Для получения более подробной

Конфигурация	Описание
	информации см. Спецификация WebAuthn .
Алгоритмы подписи	Алгоритмы, сообщающие аутентификатору WebAuthn, какие алгоритмы подписи использовать для удостоверения открытого ключа . Tuxedo SSO использует удостоверение открытого ключа для подписи и проверки утверждений аутентификации . Если алгоритмов не существует, адаптируются значения по умолчанию ES256 и RS256 . ES256 и RS256 являются необязательными элементами конфигурации, применяемыми к регистрации аутентификаторов WebAuthn. Более подробную информацию см. в спецификации WebAuthn .
Идентификатор проверяющей стороны	Идентификатор проверяющей стороны WebAuthn, определяющий область действия учетных данных открытого ключа . Идентификатор должен быть эффективным доменом источника. Этот идентификатор является необязательным элементом конфигурации, применяемым к регистрации аутентификаторов WebAuthn. Если эта запись пуста, Tuxedo SSO адаптирует хостовую часть базового URL-адреса Tuxedo SSO. Для получения более подробной информации см. Спецификацию WebAuthn .
Предпочтение при передаче удостоверения	Реализация API WebAuthn в браузере (WebAuthn Client) является предпочтительным методом для генерации утверждений об аттестации. Эта настройка является необязательным элементом конфигурации, применяемым к регистрации аутентификатора WebAuthn. Если опция не существует, ее поведение такое же, как при выборе "none". Для получения более подробной информации см. WebAuthn Specification .
Приложение-аутентификатор	Допустимый шаблон прикрепления аутентификатора WebAuthn для клиента WebAuthn. Этот шаблон является необязательным элементом конфигурации, применяемым к регистрации аутентификатора WebAuthn. Для получения более подробной информации см. Спецификация WebAuthn .
Требовать обнаруживаемые учетные данные	Параметр, требующий, чтобы аутентификатор WebAuthn генерировал Учетные данные открытого ключа как Учетные данные, обнаруживаемые на стороне клиента . Этот параметр применяется к регистрации аутентификатора WebAuthn. Если оставить пустым, его поведение будет таким же, как при выборе

Конфигурация	Описание
	"Нет". Для получения более подробной информации см. Спецификация WebAuthn .
Требование проверки пользователя	Параметр, требующий, чтобы аутентификатор WebAuthn подтвердил проверку пользователя. Это необязательный элемент конфигурации, применяемый к регистрации аутентификатора WebAuthn и аутентификации пользователя аутентификатором WebAuthn. Если параметр не указан, его поведение такое же, как при выборе «предпочтительный». Для получения более подробной информации см. Спецификация WebAuthn для регистрации аутентификатора WebAuthn и Спецификация WebAuthn для аутентификации пользователя аутентификатором WebAuthn .
Тайм-аут	Значение тайм-аута в секундах для регистрации аутентификатора WebAuthn и аутентификации пользователя с помощью аутентификатора WebAuthn. Если установлено значение ноль, его поведение зависит от реализации аутентификатора WebAuthn. Значение по умолчанию — 0. Для получения более подробной информации см. Спецификация WebAuthn для регистрации аутентификатора WebAuthn и Спецификация WebAuthn для аутентификации пользователя с помощью аутентификатора WebAuthn .
Избегайте регистрации одного и того же аутентификатора	Если этот параметр включен, Tuxedo SSO не сможет повторно зарегистрировать уже зарегистрированный аутентификатор WebAuthn.
Допустимые AAGUID	Белый список AAGUID, по которому должен регистрироваться аутентификатор WebAuthn.

Проверка заявления об аттестации

При регистрации аутентификатора WebAuthn Tuxedo SSO проверяет надежность заявления об аттестации, сгенерированного аутентификатором WebAuthn. Tuxedo SSO требует, чтобы сертификаты якоря доверия были импортированы в truststore .

Чтобы пропустить эту проверку, отключите это хранилище доверенных сертификатов или установите для элемента конфигурации политики WebAuthn «Attestation Conveyance Preference» значение «none».

Управление учетными данными WebAuthn в качестве пользователя

Регистрация аутентификатора WebAuthn
Правильный способ регистрации аутентификатора WebAuthn зависит от того, зарегистрировал ли пользователь уже учетную запись на Tuxedo SSO.

Новый пользователь
Если требуемое действие WebAuthn Register является действием по умолчанию в области, новые пользователи должны настроить пароль после своего первого входа в систему.

Процедура

1. Откройте форму входа.
2. Нажмите «Зарегистрироваться» .
3. Заполните поля формы.
4. Нажмите «Зарегистрироваться» .

После успешной регистрации браузер просит пользователя ввести текст метки своего аутентификатора WebAuthn.

Существующий пользователь
Если WebAuthn Authenticatorвсе настроено так, как показано в первом примере, то при попытке существующих пользователей войти в систему им потребуется автоматически зарегистрировать свой аутентификатор WebAuthn:

Процедура

1. Откройте форму входа.
2. Введите данные в форму.
3. Нажмите «Сохранить ».
4. Нажмите «Войти» .

После успешной регистрации браузер пользователя просит ввести текст метки его аутентификатора WebAuthn.

Беспарольная WebAuthn с двухфакторной аутентификацией

Tuxedo SSO использует WebAuthn для двухфакторной аутентификации, но вы можете использовать WebAuthn в качестве аутентификации первого фактора. В этом случае пользователи с passwordlessучетными данными WebAuthn могут аутентифицироваться в Tuxedo SSO без пароля. Tuxedo SSO может использовать WebAuthn как механизм аутентификации без пароля и двухфакторной аутентификации в контексте области и одного потока аутентификации.

Администратор обычно требует, чтобы ключи доступа, зарегистрированные пользователями для аутентификации WebAuthn без пароля, соответствовали различным требованиям. Например, ключи доступа могут требовать от пользователей аутентификации с ключом доступа с помощью PIN-кода или ключ доступа должен быть подтвержден более сильным центром сертификации.

Из-за этого Tuxedo SSO позволяет администраторам настраивать отдельный WebAuthn Passwordless Policy. Существует обязательное Webauthn Register Passwordlessдействие типа и отдельный аутентификатор типа WebAuthn Passwordless Authenticator.

Настраивать

Настройте поддержку WebAuthn без пароля следующим образом:

1. (если еще не присутствует) Зарегистрируйте новое необходимое действие для поддержки WebAuthn без пароля. Используйте шаги, описанные в разделе Включение регистрации аутентификатора WebAuthn . Зарегистрируйте Webauthn Register Passwordlessдействие.
2. Настройте политику. Вы можете использовать шаги и параметры конфигурации, описанные в разделе Управление политикой . Выполните настройку в консоли администратора на вкладке WebAuthn Passwordless Policy . Обычно требования к ключу доступа будут строже, чем для двухфакторной политики. Например, вы можете установить User Verification Requirement на Required при настройке политики без пароля.

3. Настройте поток аутентификации. Используйте поток WebAuthn Browser, описанный в разделе Добавление аутентификации WebAuthn в поток браузера . Настройте поток следующим образом:
    - Подпоток WebAuthn Browser Forms содержит Username Form в качестве первого аутентификатора. Удалите аутентификатор Username Password Form по умолчанию и добавьте аутентификатор Username Form . Это действие требует от пользователя предоставления имени пользователя в качестве первого шага.
    - Будет обязательный подпоток, который можно назвать Passwordless Or Two-factor , например. Этот подпоток указывает, что пользователь может аутентифицироваться с помощью Passwordless WebAuthn credential или с помощью Two-factor authentication.
    - В качестве первой альтернативы поток содержит WebAuthn Passwordless Authenticator .
    - Вторая альтернатива будет подпотоком с именем Password And Two-factor Webauthn , например. Этот подпоток содержит Password Form и WebAuthn Authenticator .

Окончательная конфигурация потока выглядит примерно так:

ПарольБез потока

Теперь вы можете добавить WebAuthn Register Passwordless в качестве обязательного действия для пользователя, уже известного Tuxedo SSO, чтобы протестировать это. Во время первой аутентификации пользователь должен использовать пароль и учетные данные WebAuthn второго фактора. Пользователю не нужно предоставлять пароль и учетные данные WebAuthn второго фактора, если он использует учетные данные WebAuthn Passwordless.

LoginLess WebAuthn

Tuxedo SSO использует WebAuthn для двухфакторной аутентификации, но вы можете использовать WebAuthn в качестве аутентификации первого фактора. В этом случае пользователи с passwordlessучетными данными WebAuthn могут аутентифицироваться в Tuxedo SSO без ввода логина или пароля. Tuxedo SSO

может использовать WebAuthn как механизм аутентификации без входа/пароля и двухфакторной аутентификации в контексте области.

Администратор обычно требует, чтобы ключи доступа, зарегистрированные пользователями для аутентификации без входа WebAuthn, соответствовали различным требованиям. Аутентификация без входа требует, чтобы пользователи аутентифицировались с ключом доступа (например, с помощью PIN-кода или отпечатка пальца), а криптографические ключи, связанные с учетными данными без входа, физически хранились на ключе доступа. Не все ключи доступа соответствуют таким требованиям. Уточните у поставщика ключей доступа, поддерживает ли ваше устройство «проверку пользователя» и «обнаруживаемые учетные данные». См. Поддерживаемые ключи доступа .

Tuxedo SSO позволяет администраторам настраивать WebAuthn Passwordless Policyутаким образом, чтобы разрешить аутентификацию без входа в систему. Обратите внимание, что аутентификацию без входа в систему можно настроить только с WebAuthn Passwordless Policyи с WebAuthn Passwordlessучетными данными. Аутентификация без входа в систему WebAuthn и аутентификация без пароля WebAuthn могут быть настроены в одной и той же области, но будут использовать одну и ту же политику WebAuthn Passwordless Policy.

Настраивать

Процедура

Настройте поддержку WebAuthn Loginless следующим образом:

1. (если еще не присутствует) Зарегистрируйте новое необходимое действие для поддержки WebAuthn без пароля. Используйте шаги, описанные в разделе Включение регистрации аутентификатора WebAuthn . Зарегистрируйте Webauthn Register Passwordlessдействие.
2. Настройте WebAuthn Passwordless Policy. Выполните настройку в Authenticationразделе Admin Console, на вкладке Policies→ WebAuthn Passwordless Policy. Вам необходимо установить User Verification Requirement на required и Require Discoverable Credential на Yes при настройке политики для сценария без входа. Обратите внимание, что поскольку нет специальной политики Loginless, будет невозможно

смешивать сценарии аутентификации с user verification=no/discoverable credential=no и сценариями без входа (user verification=yes/discoverable credential=yes). Емкость хранилища обычно очень ограничена для Passkeys, что означает, что вы не сможете хранить много обнаруживаемых учетных данных на своем Passkey.

3. Настройте поток аутентификации. Создайте новый поток аутентификации, добавьте выполнение "WebAuthn Passwordless" и установите параметр Requirement выполнения на Required

Окончательная конфигурация потока выглядит примерно так:

ВойтиМеньше потока

Теперь вы можете добавить требуемое действие WebAuthn Register Passwordlessдля пользователя, уже известного Tuxedo SSO, чтобы протестировать это. Пользователь с настроенным требуемым действием должен будет пройти аутентификацию (например, с помощью имени пользователя/пароля), а затем ему будет предложено зарегистрировать Passkey, который будет использоваться для аутентификации без входа в систему.

Замечания, касающиеся конкретного поставщика

Список проверки совместимости
Для аутентификации без входа в систему с помощью Tuxedo SSO требуется, чтобы ключ доступа соответствовал следующим характеристикам:

- Соответствие FIDO2: не путать с FIDO/U2F
- Проверка пользователя: возможность использования ключа доступа для аутентификации пользователя (предотвращает возможность того, что кто-то узнает ваш ключ доступа и сможет выполнить аутентификацию без входа в систему и пароля)
- Обнаруживаемые учетные данные: возможность для ключа доступа хранить логин и криптографические ключи, связанные с клиентским приложением.

Windows Привет

Чтобы использовать учетные данные на основе Windows Hello для аутентификации в Tuxedo SSO, настройте параметр Signature Algorithms для WebAuthn Passwordless Policyвключения значения RS256 . Обратите внимание, что некоторые браузеры не разрешают доступ к платформе Passkey (например, Windows Hello) внутри приватных окон.

Поддерживаемые пароли

Следующие ключи доступа были успешно протестированы для аутентификации без входа в систему с помощью Tuxedo SSO:

- Windows Hello (Windows 10 21H1/21H2)
- Юбико Юбикей 5 NFC
- Feitian ePass FIDO-NFC

Пароли

Tuxedo SSO обеспечивает предварительную поддержку Passkeys . Tuxedo SSO работает как Passkeys Relying Party (RP).

Регистрация и аутентификация ключа доступа реализованы с помощью функций WebAuthn . Таким образом, пользователи Tuxedo SSO могут выполнять регистрацию и аутентификацию ключа доступа с помощью существующей регистрации и аутентификации WebAuthn .

> Как синхронизированные ключи доступа, так и привязанные к устройству ключи доступа могут использоваться как для аутентификации на одном устройстве, так и для аутентификации между устройствами (CDA). Однако успешность операций с ключами доступа зависит от среды пользователя. Убедитесь, какие операции могут быть успешными в среде .

Аутентификация по ключу доступа с условным пользовательским интерфейсом

Аутентификация по ключу доступа с условным пользовательским интерфейсом может аутентифицировать пользователя с помощью его ключа доступа таким же образом, как в LoginLess WebAuthn . Эта аутентификация показывает пользователю список ключей доступа, хранящихся на устройстве, на котором

пользователь запускает браузер. Таким образом, пользователь может выбрать один из ключей доступа в списке для аутентификации. По сравнению с LoginLess WebAuthn , аутентификация улучшает пользовательский опыт аутентификации.

> Эта аутентификация использует WebAuthn Conditional UI . Таким образом, успех этой аутентификации зависит от среды пользователя. Если среда не поддерживает WebAuthn Conditional UI, эта аутентификация возвращается к LoginLess WebAuthn .

> Аутентификация по ключу доступа — это **предварительная версия** , которая не поддерживается полностью. Эта функция отключена по умолчанию.

> Чтобы включить запустите сервер с помощью `--features=preview`или`--features=passkeys`

Настраивать

Процедура

Настройте аутентификацию по ключу доступа с помощью условного пользовательского интерфейса следующим образом:

1. (если еще не присутствует) Зарегистрируйте новое необходимое действие для поддержки WebAuthn без пароля. Используйте шаги, описанные в разделе Включение регистрации аутентификатора WebAuthn . Зарегистрируйте Webauthn Register Passwordlessдействие.
2. Настройте WebAuthn Passwordless Policy. Выполните настройку в Authenticationразделе Admin Console, на вкладке Policies→ WebAuthn Passwordless Policy. Установите User Verification Requirement на required и Require discoverable credential на Yes при настройке политики для сценария без входа. Обратите внимание, что поскольку нет специальной политики Loginless, невозможно смешивать сценарии аутентификации с user verification=no/discoverable credential=no и сценариями без входа (user verification=yes/discoverable credential=yes).

   > Емкость хранилища обычно очень ограничена на аппаратных ключах доступа, что означает, что вы не можете хранить много обнаруживаемых учетных данных на своем ключе доступа. Однако это ограничение может быть смягчено, например, если вы используете телефон Android, поддерживаемый учетной записью Google, в качестве устройства ключа доступа или iPhone, поддерживаемый Bitwarden.

3. Настройте поток аутентификации. Создайте новый поток аутентификации, добавьте выполнение Passkeys Conditional UI Authenticator и установите настройку Requirement выполнения на Required .

   Окончательная конфигурация потока выглядит примерно так:

4. Свяжите указанный выше поток как поток аутентификации браузера в области, как описано в разделе WebAuthn выше .

Поток аутентификации выше требует, чтобы пользователь уже имел учетные данные ключа доступа в своей учетной записи, чтобы иметь возможность войти в систему. Это требование означает, что все пользователи в области должны иметь уже установленные ключи доступа. Этого можно достичь, например, включив регистрацию пользователей, как описано ниже.

Настройка регистрации для условного пользовательского интерфейса ключей доступа

1. Включить регистрацию для вашего домена
2. В потоках аутентификации области выберите регистрацию потока и переключите проверку пароля аутентификатора на Отключено . Это означает, что вновь зарегистрированным пользователям не потребуется создавать пароли в этом примере настройки. Пользователи всегда должны использовать ключи доступа вместо паролей.
3. Вернитесь на вкладку Required actions вкладки Authentication , найдите Webauthn Register Passwordlessдействие и отметьте его как Set as default action . Это означает, что оно будет добавлено ко всем новым пользователям после их регистрации.

Альтернативой настройке потока регистрации является добавление требуемого действия WebAuthn Register Passwordlessдля пользователя, который уже известен Tuxedo SSO. Пользователь с настроенным требуемым действием должен будет пройти аутентификацию (например, с помощью имени пользователя/пароля), а затем ему будет предложено зарегистрировать ключ доступа, который будет использоваться для аутентификации без входа в систему.

Мы планируем улучшить удобство использования и разрешить интеграцию условных паролей

с существующими аутентификаторами и формами, такими как форма имени пользователя и пароля по умолчанию.

На уровне 3 веб-аутентификации резидентный **ключ** был заменен на **обнаруживаемые учетные данные** .

Если браузер пользователя поддерживает условный интерфейс WebAuthn , отображается следующий экран.

Аутентификация по ключу доступа с условным пользовательским интерфейсом

Когда пользователь нажимает на текстовое поле «Выберите свой ключ доступа» , отображается список ключей доступа, хранящихся на устройстве, на котором пользователь запускает просмотр, как показано ниже.

Аутентификация по ключу доступа с условным автозаполнением пользовательского интерфейса

Если браузер пользователя не поддерживает условный пользовательский интерфейс WebAuthn , аутентификация возвращается к LoginLess WebAuthn следующим образом.

Аутентификация по ключу доступа с условным пользовательским интерфейсом, возвращающимся к LoginLess WebAuthn

Коды восстановления (RecoveryCodes)

Вы можете настроить коды восстановления для двухфакторной аутентификации, добавив «Форму кода восстановления аутентификации» в качестве двухфакторного аутентификатора в свой поток аутентификации. Пример настройки этого аутентификатора см. в WebAuthn .

RecoveryCodes — это **предварительная версия** , которая не поддерживается полностью. Эта функция отключена по умолчанию.

Чтобы включить запустите сервер с помощью `--features=preview`или`--features=recovery-codes`

Условия в условных потоках

Как упоминалось в Требованиях к выполнению , выполнения условий могут содержаться только в условном подпотоке. Если все выполнения условий оцениваются как истинные, то условный подпоток действует как обязательный . Вы можете обработать следующее выполнение в условном подпотоке. Если некоторые выполнения, включенные в условный подпоток, оцениваются как ложные, то весь подпоток считается отключенным .

Доступные условия

Condition - User Role

Это выполнение имеет возможность определить, имеет ли пользователь роль, определенную полем User role . Если у пользователя есть требуемая роль, выполнение считается истинным, и другие выполнения оцениваются. Администратор должен определить следующие поля:

Псевдоним

Описывает имя выполнения, которое будет показано в потоке аутентификации.

Роль пользователя

Роль, которую должен иметь пользователь для выполнения этого потока. Для указания роли приложения синтаксис следующий appname.approle(например myapp.myrole).

Condition - User Configured

Это проверяет, настроены ли другие исполнения в потоке для пользователя. Раздел Требования к исполнению включает пример формы OTP.

Condition - User Attribute

Это проверяет, настроил ли пользователь требуемый атрибут: опционально, проверка может также оценить атрибуты группы. Существует возможность отрицать вывод, что означает, что у пользователя не должно быть атрибута. Раздел Атрибуты пользователя показывает, как добавить пользовательский атрибут. Вы можете предоставить эти поля:

Псевдоним

Описывает имя выполнения, которое будет показано в потоке аутентификации.

Имя атрибута

Имя проверяемого атрибута.

Ожидаемое значение атрибута

Ожидаемое значение атрибута.

Включить групповые атрибуты

Если включено, условие проверяет, есть ли в какой-либо из присоединенных групп один атрибут, соответствующий настроенному имени и значению: этот параметр может повлиять на производительность.

Отрицательный вывод

Вы можете инвертировать вывод. Другими словами, атрибут не должен присутствовать.

Явно запретить/разрешить доступ в условных потоках

Вы можете разрешить или запретить доступ к ресурсам в условном потоке. Два аутентификатора Deny Accessи Allow Accessконтролируют доступ к ресурсам по условиям.

Allow Access

Аутентификатор всегда будет успешно аутентифицироваться. Этот аутентификатор не настраивается.

Deny Access

Доступ всегда будет запрещен. Вы можете определить сообщение об ошибке, которое будет показано пользователю. Вы можете предоставить следующие поля:

Псевдоним

Описывает имя выполнения, которое будет показано в потоке аутентификации.

Сообщение об ошибке

Сообщение об ошибке, которое будет показано пользователю. Сообщение об ошибке может быть предоставлено как конкретное сообщение или как свойство, чтобы использовать его с локализацией. (например, " У вас нет роли 'admin'. ", my-property-deny в свойствах сообщений) Оставьте пустым для сообщения по умолчанию, определенного как свойство access-denied.

Вот пример того, как запретить доступ всем пользователям, у которых нет роли role1, и показать сообщение об ошибке, определенное свойством deny-role1. Этот пример включает Condition - User Roleи Deny Accessвыполнения.

Поток браузера

Условие - конфигурация роли пользователя

Настройка Deny Accessочень проста. Вы можете указать произвольный псевдоним и требуемое сообщение, например:

Последнее, что нужно сделать, это определить свойство с сообщением об ошибке в теме входа messages_en.properties(для английского языка):

deny-role1 = You do not have required role!

Сеансы аутентификации

Когда страница входа открывается в первый раз в веб-браузере, Tuxedo SSO создает объект, называемый сеансом аутентификации, который хранит некоторую полезную информацию о запросе. Всякий раз, когда новая страница входа открывается с другой вкладки в том же браузере, Tuxedo SSO создает новую запись, называемую подсеансом аутентификации, которая сохраняется в сеансе аутентификации. Запросы аутентификации могут поступать от любого типа клиентов, например, Admin CLI. В этом случае также создается новый сеанс аутентификации с одним подсеансом аутентификации. Обратите внимание, что сеансы аутентификации можно создавать и другими способами, помимо использования потока браузера.

Сеанс аутентификации обычно истекает через 30 минут по умолчанию. Точное время указывается переключателем Login timeout на вкладке Sessions консоли администратора при настройке областей .

Аутентификация в большем количестве вкладок браузера

Как описано в предыдущем разделе, ситуация может включать пользователя, который пытается пройти аутентификацию на сервере Tuxedo SSO с нескольких вкладок одного браузера. Однако, когда этот пользователь проходит аутентификацию на одной вкладке браузера, другие вкладки браузера автоматически перезапускают аутентификацию. Эта аутентификация происходит из-за небольшого javascript, доступного на страницах входа Tuxedo SSO. Перезапуск обычно аутентифицирует пользователя на других вкладках браузера и перенаправляет на клиентов, поскольку сейчас есть сеанс SSO из-за того, что пользователь только что успешно прошел аутентификацию на первой вкладке браузера. Существуют некоторые редкие исключения, когда пользователь не проходит автоматическую аутентификацию на других вкладках браузера, например, при использовании параметра OIDC prompt=login или аутентификация пошагово, запрашивающая более сильный фактор аутентификации, чем текущий аутентифицированный фактор.

В некоторых редких случаях может случиться так, что после аутентификации на первой вкладке браузера другие вкладки браузера не смогут перезапустить аутентификацию, поскольку сеанс аутентификации уже истек. В этом случае конкретная вкладка браузера перенаправит ошибку об истекшем сеансе аутентификации обратно клиенту специфическим для протокола способом. Более подробную информацию см. в соответствующих разделах документации OIDC в разделе «Безопасные приложения» . Когда клиентское приложение получает такую ошибку, оно может немедленно повторно отправить запрос аутентификации OIDC/SAML в Tuxedo SSO, поскольку это обычно должно автоматически аутентифицировать пользователя из-за существующего сеанса SSO, как описано ранее. В результате конечный пользователь автоматически аутентифицируется на всех вкладках браузера. Адаптер JavaScript Tuxedo SSO в разделе «Безопасные приложения» и поставщик удостоверений Tuxedo SSO поддерживают автоматическую обработку этой ошибки и повторную попытку аутентификации на сервере Tuxedo SSO в таком случае.

Интеграция поставщиков удостоверений

Брокер идентификации — это посредническая служба, соединяющая поставщиков услуг с поставщиками идентификации. Брокер идентификации создает связь с внешним поставщиком идентификации, чтобы использовать идентификаторы поставщика для доступа к внутренним услугам, которые предоставляет поставщик услуг.

С точки зрения пользователя брокеры идентификации предоставляют ориентированный на пользователя централизованный способ управления идентификациями для доменов и областей безопасности. Вы можете связать учетную запись с одной или несколькими идентификациями от поставщиков идентификации или создать учетную запись на основе информации об идентификации от них.

Поставщик идентификации происходит от определенного протокола, используемого для аутентификации и отправки пользователям информации об аутентификации и авторизации. Это может быть:

- Социальный провайдер, такой как Facebook, Google или Twitter.
- Деловой партнер, пользователям которого необходим доступ к вашим услугам.
- Облачный сервис идентификации, который вы хотите интегрировать.

Обычно Tuxedo SSO основывает поставщиков удостоверений на следующих протоколах:

- SAML v2.0
- OpenID Connect v1.0
- OAuth v2.0

Обзор брокерской деятельности

При использовании Tuxedo SSO в качестве брокера идентификации, Tuxedo SSO не заставляет пользователей предоставлять свои учетные данные для аутентификации в определенной области. Tuxedo SSO отображает список поставщиков идентификации, у которых они могут пройти аутентификацию.

Если вы настроите поставщика удостоверений по умолчанию, Tuxedo SSO перенаправит пользователей к поставщику по умолчанию.

Различные протоколы могут требовать различных потоков аутентификации. Все поставщики удостоверений, поддерживаемые Tuxedo SSO, используют следующий поток.

Поток брокера идентичности

1. Неаутентифицированный пользователь запрашивает защищенный ресурс в клиентском приложении.
2. Клиентское приложение перенаправляет пользователя в Tuxedo SSO для аутентификации.
3. Tuxedo SSO отображает страницу входа со списком поставщиков удостоверений, настроенных в области.
4. Пользователь выбирает одного из поставщиков удостоверений, нажимая на его кнопку или ссылку.

5. Tuxedo SSO отправляет запрос на аутентификацию целевому поставщику удостоверений, запрашивающему аутентификацию, и перенаправляет пользователя на страницу входа поставщика удостоверений. Администратор уже установил свойства соединения и другие параметры конфигурации для поставщика удостоверений консоли администратора.
6. Пользователь предоставляет учетные данные или дает согласие на аутентификацию у поставщика удостоверений.
7. После успешной аутентификации провайдером идентификации пользователь перенаправляется обратно в Tuxedo SSO с ответом аутентификации. Обычно ответ содержит токен безопасности, используемый Tuxedo SSO для доверия аутентификации провайдера идентификации и получения информации о пользователе.
8. Tuxedo SSO проверяет, является ли ответ от поставщика удостоверений действительным. Если ответ действителен, Tuxedo SSO импортирует и создает пользователя, если пользователь еще не существует. Tuxedo SSO может запросить у поставщика удостоверений дополнительную информацию о пользователе, если токен не содержит этой информации. Такое поведение называется федерацией удостоверений . Если пользователь уже существует, Tuxedo SSO может попросить пользователя связать удостоверение, возвращенное поставщиком удостоверений, с существующей учетной записью. Такое поведение называется привязкой учетных записей . С помощью Tuxedo SSO вы можете настроить привязку учетных записей и указать ее в потоке первого входа . На этом этапе Tuxedo SSO аутентифицирует пользователя и выдает его токен для доступа к запрошенному ресурсу у поставщика услуг.
9. Когда пользователь проходит аутентификацию, Tuxedo SSO перенаправляет пользователя к поставщику услуг, отправляя токен, ранее выданный во время локальной аутентификации.
10. Поставщик услуг получает токен от Tuxedo SSO и разрешает доступ к защищенному ресурсу.

Возможны вариации этого потока. Например, клиентское приложение может запросить определенного поставщика удостоверений, а не отображать их список, или вы можете настроить Tuxedo SSO так, чтобы заставить пользователей предоставить дополнительную информацию перед объединением их удостоверений.

В конце процесса аутентификации Tuxedo SSO выдает свой токен клиентским приложениям. Клиентские приложения отделены от внешних поставщиков удостоверений, поэтому они не могут видеть протокол клиентского приложения или то, как они проверяют личность пользователя. Поставщику нужно знать только о Tuxedo SSO.

Поставщик удостоверений по умолчанию

Tuxedo SSO может перенаправлять к поставщику удостоверений вместо отображения формы входа. Чтобы включить это перенаправление:

Процедура

1. Нажмите «Аутентификация» в меню.
2. Нажмите на ссылку «Браузер» .
3. Нажмите на значок шестеренки ⚙ в строке «Перенаправление поставщика удостоверений» .
4. Установите в качестве поставщика удостоверений по умолчанию поставщика удостоверений, к которому вы хотите перенаправлять пользователей.

Если Tuxedo SSO не находит настроенного поставщика удостоверений по умолчанию, отображается форма входа.

Этот аутентификатор отвечает за обработку kc_idp_hintпараметра запроса. Для получения дополнительной информации см. раздел «Предложенный клиентом поставщик удостоверений» .

Общая конфигурация

Основой конфигурации брокера идентификации являются поставщики идентификации (IDP). Tuxedo SSO создает поставщиков идентификации для каждой области и включает их для каждого приложения по умолчанию. Пользователи из области могут использовать любого из зарегистрированных поставщиков идентификации при входе в приложение.

Процедура

1. В меню выберите «Поставщики удостоверений» .

   Поставщики удостоверений личности

2. Выберите поставщика удостоверений. Tuxedo SSO отображает страницу конфигурации для выбранного вами поставщика удостоверений.

   Добавить поставщика удостоверений Facebook

   При настройке поставщика удостоверений поставщик удостоверений отображается на странице входа Tuxedo SSO в качестве опции. Вы можете разместить пользовательские значки на экране входа для каждого поставщика удостоверений. См. пользовательские значки для получения дополнительной информации.

   Страница входа в систему IDP

   Социальный

   Социальные провайдеры обеспечивают социальную аутентификацию в вашей области. С Tuxedo SSO пользователи могут входить в ваше приложение, используя учетную запись социальной сети. Поддерживаемые провайдеры включают Twitter, Facebook, Google, LinkedIn, Instagram, Microsoft, PayPal, Openshift v3, GitHub, GitLab, Bitbucket и Stack Overflow.

   На основе протокола

   Поставщики на основе протоколов полагаются на определенные протоколы для аутентификации и авторизации пользователей.

Используя этих поставщиков, вы можете подключиться к любому поставщику удостоверений, совместимому с определенным протоколом. Tuxedo SSO обеспечивает поддержку протоколов SAML v2.0 и OpenID Connect v1.0. Вы можете настроить и выступить в качестве посредника любого поставщика удостоверений на основе этих открытых стандартов.

Хотя каждый тип поставщика удостоверений имеет свои параметры конфигурации, все они имеют общую конфигурацию. Доступны следующие параметры конфигурации:

Таблица 1. Общая конфигурация

Конфигурация	Описание
Псевдоним	Псевдоним — это уникальный идентификатор поставщика удостоверений, который ссылается на внутреннего поставщика удостоверений. Tuxedo SSO использует псевдоним для создания URI перенаправления для протоколов OpenID Connect, которым требуется URI перенаправления или URL обратного вызова для связи с поставщиком удостоверений. Все поставщики удостоверений должны иметь псевдоним. Примеры псевдонимов включают `facebook`, `google`и `idp.acme.com`.
Включено	Включает или выключает провайдера.
Скрыть на странице входа	Когда **ON** , Tuxedo SSO не отображает этого провайдера как вариант входа на странице входа. Клиенты могут запросить этого провайдера, используя параметр 'kc_idp_hint' в URL для запроса входа.
Только привязка аккаунта	Когда **ON** , Tuxedo SSO связывает существующие учетные записи с этим провайдером. Этот провайдер не может вводить пользователей в систему, и Tuxedo SSO не отображает этого провайдера в качестве опции на странице входа.
Хранить токены	При **включении** Tuxedo SSO сохраняет токены от поставщика удостоверений.
Сохраненные токены читаемы	Когда **ON** , пользователи могут извлечь сохраненный токен поставщика удостоверений. Это действие также применяется к *брокерской* роли уровня клиента *read token* .
Доверие к электронной	Когда **ON** , Tuxedo SSO доверяет адресам электронной почты от поставщика удостоверений. Если область требует проверки электронной почты,

Конфигурация	Описание
почте	пользователям, которые входят в систему от этого поставщика удостоверений, не нужно выполнять процесс проверки электронной почты.
Заказ графического интерфейса	Порядок сортировки доступных поставщиков удостоверений на странице входа.
Проверить существенное требование	Если **включено** , идентификационные токены, выданные поставщиком удостоверений, должны иметь определенное требование, в противном случае пользователь не сможет пройти аутентификацию через этого брокера.
Основное требование	Если **проверка основного утверждения** включена , имя утверждения токена JWT для фильтрации (соответствие чувствительно к регистру **)**
Существенная стоимость иска	Если **проверка основного утверждения** включена , значение утверждения токена JWT будет соответствовать (поддерживает формат регулярных выражений **)**
Первый процесс входа в систему	Поток аутентификации Tuxedo SSO запускается, когда пользователи используют этого поставщика удостоверений для первого входа в Tuxedo SSO.
Поток входа в систему	Поток аутентификации Tuxedo SSO запускается, когда пользователь завершает вход в систему с помощью внешнего поставщика удостоверений.
Режим синхронизации	Стратегия обновления информации о пользователе от поставщика удостоверений через сопоставители. При выборе **устаревшего** , Tuxedo SSO использовал текущее поведение. **Импорт** не обновляет данные пользователя и **принудительно** обновляет данные пользователя, когда это возможно. См. Сопоставители поставщика удостоверений для получения дополнительной информации.
Имя пользователя чувствительно к регистру	Если включено, исходное имя пользователя от поставщика удостоверений сохраняется как есть при объединении пользователей. В противном случае имя пользователя от поставщика удостоверений будет в нижнем регистре и может не совпадать с исходным значением, если оно чувствительно к регистру. Этот параметр влияет только на имя пользователя, связанное с объединенным удостоверением, поскольку имена пользователей на сервере всегда в нижнем регистре.

Поставщики социальной идентичности

Поставщик социальной идентификации может делегировать аутентификацию доверенному, уважаемому аккаунту социальной сети. Tuxedo SSO включает поддержку социальных сетей, таких как Google, Facebook, Twitter, GitHub, LinkedIn, Microsoft и Stack Overflow.

Битбакет

Чтобы войти в систему Bitbucket, выполните следующую процедуру.

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Bitbucket .

   Добавить поставщика удостоверений
3. Скопируйте значение Redirect URI в буфер обмена.
4. В отдельной вкладке браузера выполните процесс OAuth на Bitbucket Cloud . Когда вы нажмете Add Consumer :
   a. Вставьте значение URI перенаправления в поле URL обратного вызова .
   b. Убедитесь, что вы выбрали «Электронная почта» и «Чтение» в разделе «Учетная запись» , чтобы разрешить приложению читать электронную почту.
5. Обратите внимание на значения Key и Secret , которые Bitbucket отображает при создании потребителя.
6. В Tuxedo SSO вставьте значение Keyв поле Client ID .
7. В Tuxedo SSO вставьте значение Secretв поле Client Secret .
8. Нажмите Добавить .

Фейсбук

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Facebook .

   Добавить поставщика удостоверений
3. Скопируйте значение Redirect URI в буфер обмена.
4. В отдельной вкладке браузера откройте Meta for Developers .
   a. Нажмите Мои приложения .
   b. Выберите Создать приложение .

      Добавить вариант использования
   c. Выберите Другое .

      Выберите тип приложения
   d. Выберите Потребитель .

      Создать приложение
   e. Заполните все обязательные поля.
   f. Нажмите «Создать приложение» . Meta перенесет вас на панель управления.

      Добавить продукт
   g. Нажмите «Настроить» в поле «Вход через Facebook» .
   h. Выберите Интернет .
   i. Введите значение URI перенаправления в поле URL-адрес сайта и нажмите Сохранить .
   j. На панели навигации выберите Настройки приложения - Основные .
   k. Нажмите Показать в поле Секрет приложения .

l. Запишите идентификатор приложения и секрет приложения .

5. Введите значения App IDиApp Secret из вашего приложения Facebook в поля «Идентификатор клиента» и «Секретный код клиента» в Tuxedo SSO.
6. Нажмите «Добавить».
7. Введите требуемые области в поле Default Scopes . По умолчанию Tuxedo SSO использует область email . Подробнее об областях Facebook см. в Graph API .

Tuxedo SSO отправляет запросы профиля graph.facebook.com/me?fields=id,name,email,first_name,last_nameпо умолчанию. Ответ содержит только поля id, name, email, first_name и last_name. Чтобы получить дополнительные поля из профиля Facebook, добавьте соответствующую область действия и имя поля в Additional user’s profile fieldsполе параметра конфигурации.

GitHub

Чтобы войти в GitHub, выполните следующую процедуру.

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Github .

   Добавить поставщика удостоверений
3. Скопируйте значение Redirect URI в буфер обмена.
4. В отдельной вкладке браузера создайте приложение OAUTH .
   a. Введите значение Redirect URI в поле Authorization callback URL при создании приложения.
   b. Запишите идентификатор клиента и секретный ключ клиента на странице управления вашего приложения OAUTH.
5. В Tuxedo SSO вставьте значение Client IDв поле Client ID .
6. В Tuxedo SSO вставьте значение Client secretв поле Client Secret .

7. Нажмите Добавить .

GitLab

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке Добавить поставщика выберите GitLab .

   Добавить поставщика удостоверений
3. Скопируйте значение Redirect URI в буфер обмена.
4. В отдельной вкладке браузера добавьте новое приложение GitLab .
   a. Используйте URI перенаправления в буфере обмена в качестве URI перенаправления .
   b. Запишите идентификатор приложения и секретный ключ при сохранении приложения.
5. В Tuxedo SSO вставьте значение Application IDв поле Client ID .
6. В Tuxedo SSO вставьте значение Secretв поле Client Secret .
7. Нажмите Добавить .

Google

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Google .

   Добавить поставщика удостоверений
3. Скопируйте значение Redirect URI в буфер обмена.
4. В отдельной вкладке браузера откройте консоль Google Cloud Platform .

5. В панели инструментов Google для вашего приложения Google, в меню навигации слева, наведите курсор на API и службы , а затем нажмите на опцию экрана согласия OAuth . Создайте экран согласия, убедившись, что тип пользователя экрана согласия — Внешний .
6. В панели инструментов Google:
    a. Нажмите меню «Учетные данные» .
    b. Нажмите СОЗДАТЬ УЧЕТНЫЕ ДАННЫЕ - OAuth Client ID .
    c. В списке Тип приложения выберите Веб-приложение .
    d. Используйте URI перенаправления в буфере обмена в качестве разрешенных URI перенаправления.
    e. Нажмите «Создать» .
    f. Запишите свой идентификатор клиента и свой секретный код клиента .
7. В Tuxedo SSO вставьте значение Your Client IDв поле Client ID .
8. В Tuxedo SSO вставьте значение Your Client secretв поле Client Secret .
9. Нажмите «Добавить».
10. Введите требуемые области в поле Default Scopes . По умолчанию Tuxedo SSO использует следующие области: openid profile email . Список областей Google см. в OAuth Playground .
11. Чтобы ограничить доступ только для членов вашей организации GSuite, введите домен G Suite в поле Размещенный домен .
12. Нажмите «Сохранить ».

Инстаграм

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Instagram .

Добавить поставщика удостоверений

3. Скопируйте значение Redirect URI в буфер обмена.
4. В отдельной вкладке браузера откройте Meta for Developers .
    a. Нажмите Мои приложения .
    b. Выберите Создать приложение .

       Добавить вариант использования
    c. Выберите Другое .

       Выберите тип приложения
    d. Выберите Потребитель .

       Создать приложение
    e. Заполните все обязательные поля.
    f. Нажмите «Создать приложение» . Meta перенесет вас на панель управления.
    g. На панели навигации выберите Настройки приложения - Основные .
    h. Выберите + Добавить платформу внизу страницы.
    i. Нажмите [Веб-сайт] .
    j. Введите URL вашего сайта.

       Добавить продукт
    k. Выберите в меню пункт Панель управления .
    l. Нажмите «Настроить» в поле «Базовый дисплей Instagram» .
    m. Нажмите «Создать новое приложение» .

       Создайте новый идентификатор приложения Instagram
    n. Введите значение в поле Отображаемое имя .

       Настройте приложение

o. Вставьте URL-адрес перенаправления из Tuxedo SSO в поле Допустимые URI перенаправления OAuth .

p. Вставьте URL-адрес перенаправления из Tuxedo SSO в поле URL-адрес обратного вызова для деавторизации .

q. Вставьте URL-адрес перенаправления из Tuxedo SSO в поле URL-адрес запроса на удаление данных .

r. Нажмите «Показать» в поле «Секрет приложения Instagram» .

s. Запишите идентификатор приложения Instagram и секретный ключ приложения Instagram .

t. Нажмите «Обзор приложения» — «Запросы» .

u. Следуйте инструкциям на экране.

5. В Tuxedo SSO вставьте значение Instagram App IDв поле Client ID .
6. В Tuxedo SSO вставьте значение Instagram App Secretв поле Client Secret .
7. Нажмите Добавить .

LinkedIn

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите LinkedIn .

   Добавить поставщика удостоверений

3. Скопируйте значение Redirect URI в буфер обмена.
4. В отдельной вкладке браузера создайте приложение на портале разработчиков LinkedIn.

   a. После создания приложения нажмите вкладку «Аутентификация» .

   b. Введите значение URI перенаправления в поле Разрешенные URL-адреса перенаправления для вашего приложения .

c. Запишите свой идентификатор клиента и свой секретный код клиента .

d. Откройте вкладку «Продукты» и запросите доступ для входа в LinkedIn с использованием продукта OpenID Connect.

5. В Tuxedo SSO вставьте значение Client IDв поле Client ID .
6. В Tuxedo SSO вставьте значение Client Secretв поле Client Secret .
7. Нажмите Добавить .

Майкрософт

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Microsoft .

   Добавить поставщика удостоверений
3. Скопируйте значение Redirect URI в буфер обмена.
4. На отдельной вкладке браузера зарегистрируйте приложение в Microsoft Azure в разделе «Регистрация приложений» .

   a. В разделе Redirect URI выберите Web в качестве платформы и вставьте значение Redirect URI в поле.

   b. Найдите свое приложение в разделе «Регистрация приложений» и добавьте новый секретный ключ клиента в разделе «Сертификаты и секреты» .

   c. Обратите внимание на ценность созданного секрета.

   d. Обратите внимание на идентификатор приложения (клиента) в разделе «Обзор» .
5. В Tuxedo SSO вставьте значение Application (client) IDв поле Client ID .
6. В Tuxedo SSO вставьте Valuecекретный ключ в поле «Секрет клиента» .
7. Нажмите Добавить .

OpenShift3

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Openshift v3 .

   Добавить поставщика удостоверений
3. Скопируйте значение Redirect URI в буфер обмена.
4. Зарегистрируйте своего клиента с помощью осинструмента командной строки.

```
$ oc create -f <(echo '
kind: OAuthClient
apiVersion: v1
metadata:
  name: kc-client (1)
secret: "..." (2)
redirectURIs:
  - "http://www.example.com/" (3)
grantMethod: prompt (4)
')
```

1 Вашего `name`клиента OAuth. Передается как `client_id`параметр запроса при выполнении запросов к *`<openshift_master>`*`/oauth/authorize`и *`<openshift_master>`*`/oauth/token`.

2 Tuxedo SSO `secret`использует для `client_secret`параметра запроса.

3 Параметр `redirect_uri`, указанный в запросах к *`<openshift_master>`*`/oauth/authorize`и *`<openshift_master>`*`/oauth/token` должен быть равен (или иметь префикс) одному из URI в `redirectURIs`. Вы можете получить его из поля **Redirect URI** на экране Identity Provider

4 Tuxedo SSO `grantMethod`используется для определения действия, когда этот клиент запрашивает токены, но не получил доступ от пользователя.

1. В Tuxedo SSO вставьте значение **идентификатора клиента** в поле **«Идентификатор клиента»** .

2. В Tuxedo SSO вставьте значение **Client Secret** в поле **Client Secret** .
3. Нажмите **Добавить** .

OpenShift 4

Предпосылки

1. Сертификат экземпляра OpenShift 4, хранящийся в хранилище Tuxedo SSO Truststore.
2. Сервер Tuxedo SSO, настроенный для использования truststore. Для получения дополнительной информации см. руководство Configuring a Truststore .

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Openshift v4 .
3. Введите идентификатор клиента и секрет клиента , а в поле базового URL введите URL-адрес API вашего экземпляра OpenShift 4. Кроме того, вы можете скопировать URI перенаправления в буфер обмена.

   Добавить поставщика удостоверений
4. Зарегистрируйте своего клиента либо через консоль OpenShift 4 (Главная → API Explorer → Клиент OAuth → Экземпляры), либо с помощью осинструмента командной строки.

   ```
   $ oc create -f <(echo '
   kind: OAuthClient
   apiVersion: oauth.openshift.io/v1
   metadata:
     name: kc-client (1)
   secret: "..." (2)
   redirectURIs:
     - "<here you can paste the Redirect URI that you copied in the previous step>"
   (3)
   grantMethod: prompt (4)
   ```

')

1 Вашего `name`клиента OAuth. Передается как `client_id`параметр запроса при выполнении запросов к `<openshift_master>/oauth/authorize`и `<openshift_master>/oauth/token`. `name`Параметр должен быть одинаковым в `OAuthClient`объекте и конфигурации Tuxedo SSO.

2 Tuxedo SSO `secret`использует в качестве `client_secret`параметра запроса.

3 Параметр `redirect_uri`, указанный в запросах к `<openshift_master>/oauth/authorize`и `<openshift_master>/oauth/token` должен быть равен (или иметь префикс) одному из URI в `redirectURIs`. Самый простой способ правильно настроить его — скопировать и вставить его со страницы конфигурации поставщика удостоверений Tuxedo SSO OpenShift 4 (`Redirect URI`поле).

4 Tuxedo SSO `grantMethod`используется для определения действия, когда этот клиент запрашивает токены, но не получил доступ от пользователя.

В конце концов вы должны увидеть OpenShift 4 Identity Provider на странице входа вашего экземпляра Tuxedo SSO. После нажатия на него вы должны быть перенаправлены на страницу входа OpenShift 4.

Результат

Более подробную информацию смотрите в официальной документации OpenShift .

PayPal

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите PayPal .

   Добавить поставщика удостоверений
3. Скопируйте значение Redirect URI в буфер обмена.
4. В отдельной вкладке браузера откройте область приложений PayPal Developer .
   a. Нажмите «Создать приложение» , чтобы создать приложение PayPal.
   b. Запишите идентификатор клиента и секрет клиента . Нажмите ссылку «Показать» , чтобы просмотреть секрет.

c. Убедитесь, что установлен флажок «Войти через PayPal» .

d. В разделе «Войти через PayPal» нажмите «Дополнительные настройки» .

e. Установите значение поля Return URL на значение Redirect URI from Tuxedo SSO. Обратите внимание, что URL не может содержать localhost. Если вы хотите использовать Tuxedo SSO локально, замените localhostв Return URL на , 127.0.0.1а затем получите доступ к Tuxedo SSO, используя 127.0.0.1в браузере вместо localhost.

f. Убедитесь, что поля «Полное имя» и «Электронная почта» отмечены галочками.

g. Нажмите «Сохранить» , а затем «Сохранить изменения» .

5. В Tuxedo SSO вставьте значение Client IDв поле Client ID .
6. В Tuxedo SSO вставьте значение Secret key 1в поле Client Secret .
7. Нажмите Добавить .

Переполнение стека

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Stack Overflow .

   Добавить поставщика удостоверений
3. В отдельной вкладке браузера войдите в систему регистрации своего приложения на Stack Apps .

   Зарегистрировать заявку

   a. Введите название вашего приложения в поле «Название приложения» .

   b. Введите домен OAuth в поле «Домен OAuth» .

   c. Нажмите «Зарегистрировать заявку» .

Настройки

4. Запишите идентификатор клиента , секрет клиента и ключ .
5. В Tuxedo SSO вставьте значение Client Idв поле Client ID .
6. В Tuxedo SSO вставьте значение Client Secretв поле Client Secret .
7. В Tuxedo SSO вставьте значение Keyв поле Key .
8. Нажмите Добавить .

Твиттер

Предпосылки

1. Аккаунт разработчика в Twitter.

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. В списке «Добавить поставщика» выберите Twitter .

   Добавить поставщика удостоверений
3. Скопируйте значение Redirect URI в буфер обмена.
4. В отдельной вкладке браузера создайте приложение в Twitter Application Management .
   a. Введите имя приложения и нажмите «Далее» .
   b. Запишите значения API Key и API Key Secret и нажмите Настройки приложения .
   c. В разделе Настройки аутентификации пользователя нажмите кнопку Настроить .
   d. Выберите Веб-приложение в качестве типа приложения .
   e. Вставьте значение URL-адреса перенаправления в поле URI обратного вызова / URL-адрес перенаправления .

f. Значение URL-адреса веб-сайта может быть любым допустимым URL-адресом, кроме localhost.

g. Нажмите «Сохранить» , а затем «Готово» .

5. В Tuxedo SSO вставьте значение API Keyв поле Client ID .
6. В Tuxedo SSO вставьте значение API Key Secretв поле Client Secret .
7. Нажмите Добавить .

Поставщики удостоверений OpenID Connect v1.0

Tuxedo SSO выступает в роли брокера поставщиков удостоверений на основе протокола OpenID Connect. Эти поставщики удостоверений (IDP) должны поддерживать поток кода авторизации , определенный в спецификации, для аутентификации пользователей и авторизации доступа.

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. Из Add providerсписка выберите OpenID Connect v1.0.

   Добавить поставщика удостоверений

3. Введите начальные параметры конфигурации. Подробнее о параметрах конфигурации см. в разделе Общая конфигурация IDP .

Таблица 2. Конфигурация подключения OpenID

Конфигурация	Описание
URL-адрес авторизации	Конечная точка URL-адреса авторизации, требуемая протоколом OIDC.
URL-адрес токена	Конечная точка URL-адреса токена, требуемая протоколом OIDC.
URL-адрес выхода из системы	Конечная точка URL выхода из системы в протоколе OIDC. Это значение необязательно.
Выход из обратного канала	Фоновый, внеполосный, REST-запрос к IDP для выхода пользователя из системы. Некоторые IDP выполняют выход только через

Конфигурация	Описание
	перенаправления браузера, поскольку они могут идентифицировать сеансы с помощью cookie-файла браузера.
URL-адрес информации о пользователе	Конечная точка, определяемая протоколом OIDC. Эта конечная точка указывает на информацию о профиле пользователя.
Аутентификация клиента	Определяет метод аутентификации клиента, который Tuxedo SSO использует с потоком кода авторизации. В случае JWT, подписанного закрытым ключом, Tuxedo SSO использует закрытый ключ области. В других случаях определите секрет клиента. Для получения дополнительной информации см. спецификации аутентификации клиента .
Идентификатор клиента	Область, действующая как клиент OIDC для внешнего IDP. Область должна иметь идентификатор клиента OIDC, если вы используете поток кода авторизации для взаимодействия с внешним IDP.
Секрет клиента	Секрет клиента из внешнего хранилища . Этот секрет необходим, если вы используете поток кода авторизации.
Алгоритм подписи утверждения клиента	Алгоритм подписи для создания утверждения JWT в качестве аутентификации клиента. В случае JWT, подписанного с помощью закрытого ключа или клиентского секрета как jwt, он обязателен. Если алгоритм не указан, адаптируется следующий алгоритм. `RS256`адаптируется в случае JWT, подписанного с помощью закрытого ключа. `HS256`адаптируется в случае клиентского секрета как jwt.
Аудитория утверждения клиента	Аудитория для использования в утверждении клиента. Значение по умолчанию — URL конечной точки токена IDP.
Эмитент	Tuxedo SSO проверяет утверждения эмитента в ответах от IDP на соответствие этому значению.
Области действия по умолчанию	Список областей OIDC, которые Tuxedo SSO отправляет с запросом аутентификации. Значение по умолчанию — `openid`. Каждая область разделяется пробелом.
Быстрый	Параметр prompt в спецификации OIDC. С помощью этого параметра

Конфигурация	Описание
	можно принудительно выполнить повторную аутентификацию и другие параметры. Подробнее см. в спецификации.
Принимает prompt=none пересылать от клиента	Указывает, принимает ли IDP пересылаемые запросы аутентификации, содержащие `prompt=none`параметр запроса. Если область получает запрос аутентификации с `prompt=none`, область проверяет, аутентифицирован ли пользователь в данный момент, и возвращает `login_required`ошибку, если пользователь не вошел в систему. Когда Tuxedo SSO определяет IDP по умолчанию для запроса аутентификации (используя `kc_idp_hint`параметр запроса или имея IDP по умолчанию для области), вы можете переслать запрос аутентификации с `prompt=none`на IDP по умолчанию. IDP по умолчанию проверяет аутентификацию пользователя там. Поскольку не все IDP поддерживают запросы с `prompt=none`, Tuxedo SSO использует этот переключатель, чтобы указать, что IDP по умолчанию поддерживает параметр, перед перенаправлением запроса аутентификации. Если пользователь не аутентифицирован в IDP, клиент все равно получает `login_required`ошибку. Если пользователь аутентифицирован в IDP, клиент все равно может получить `interaction_required`ошибку, если Tuxedo SSO должен отображать страницы аутентификации, требующие взаимодействия с пользователем. Эта аутентификация включает требуемые действия (например, изменение пароля), экраны согласия и экраны, настроенные для отображения потоком `first broker login`или `post broker login`потоком.
Проверить подписи	Указывает, проверяет ли Tuxedo SSO подписи на внешнем токене ID, подписанном этим IDP. Если **ON** , Tuxedo SSO должен знать открытый ключ внешнего OIDC IDP. В целях повышения производительности Tuxedo SSO кэширует открытый ключ внешнего поставщика идентификации OIDC.
Использовать URL-адрес JWKS	Этот переключатель применим, если `Validate Signatures`ON **.** Если **Use JWKS URL** ON **,** Tuxedo SSO загружает открытые ключи IDP с JWKS URL. Новые ключи загружаются, когда поставщик удостоверений генерирует новую пару ключей. Если **OFF** , Tuxedo SSO использует открытый ключ (или сертификат) из своей базы данных, поэтому при изменении пары ключей IDP импортируйте новый ключ в базу данных Tuxedo SSO.

Конфигурация	Описание
URL-адрес JWKS	URL, указывающий на местоположение ключей IDP JWK. Для получения дополнительной информации см. спецификацию JWK . Если вы используете внешний Tuxedo SSO в качестве IDP, вы можете использовать URL, такой как http://broker-Tuxedo SSO:8180/realms/test/protocol/openid-connect/certs, если ваш посреднический Tuxedo SSO работает на http://broker-Tuxedo SSO:8180 и его область — `test`.
Проверка открытого ключа	Открытый ключ в формате PEM, который Tuxedo SSO использует для проверки внешних подписей IDP. Этот ключ применяется, `Use JWKS URL`если **OFF** .
Проверка идентификатора открытого ключа	Этот параметр применяется, если **Use JWKS URL** имеет **значение OFF** . Этот параметр указывает идентификатор открытого ключа в формате PEM. Поскольку не существует стандартного способа вычисления идентификатора ключа из ключа, внешние поставщики удостоверений могут использовать алгоритмы, отличные от тех, которые использует Tuxedo SSO. Если значение этого поля не указано, Tuxedo SSO использует проверочный открытый ключ для всех запросов, независимо от идентификатора ключа, отправленного внешним IDP. Если значение **ON** , значение этого поля представляет собой идентификатор ключа, используемый Tuxedo SSO для проверки подписей от поставщиков, и должен соответствовать идентификатору ключа, указанному IDP.

Вы можете импортировать все эти данные конфигурации, указав URL или файл, указывающий на метаданные поставщика OpenID. Если вы подключаетесь к внешнему IDP Tuxedo SSO, вы можете импортировать настройки IDP из <root>/realms/{realm-name}/.well-known/openid-configuration. Эта ссылка представляет собой документ JSON, описывающий метаданные об IDP.

Если вы хотите использовать токены Json Web Encryption (JWE) ID или ответы UserInfo в провайдере, IDP должен знать открытый ключ для использования с Tuxedo SSO. Провайдер использует ключи области , определенные для различных алгоритмов шифрования, для расшифровки токенов. Tuxedo SSO предоставляет стандартную конечную точку JWKS , которую IDP может использовать для автоматической загрузки ключей.

Поставщики удостоверений SAML v2.0

Tuxedo SSO может выступать посредником между поставщиками удостоверений на основе протокола SAML v2.0.

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. Из Add providerсписка выберите SAML v2.0.

   Добавить поставщика удостоверений

3. Введите начальные параметры конфигурации. Подробнее о параметрах конфигурации см. в разделе Общая конфигурация IDP .

Таблица 3. Конфигурация SAML

Конфигурация	Описание
Идентификатор поставщика услуг	Идентификатор сущности SAML, который удаленный поставщик удостоверений использует для идентификации запросов от этого поставщика услуг. По умолчанию этот параметр установлен на URL-адрес базы областей `<root>/realms/{realm-name}`.
Идентификатор сущности поставщика удостоверений	Идентификатор сущности, используемый для проверки эмитента для полученных утверждений SAML. Если пусто, проверка эмитента не выполняется.
URL-адрес службы единого входа	Конечная точка SAML, которая запускает процесс аутентификации. Если ваш SAML IDP публикует дескриптор сущности IDP, значение этого поля указывается там.
URL-адрес службы артефактов	Конечная точка разрешения артефакта SAML. Если ваш SAML IDP публикует дескриптор сущности IDP, значение этого поля указывается там.
URL-адрес службы единого выхода	Конечная точка выхода из системы SAML. Если ваш SAML IDP публикует дескриптор сущности IDP, значение этого поля указывается там.

Конфигурация	Описание
Выход из обратного канала	Установите этот переключатель в **положение «ВКЛ»** , если ваш SAML IDP поддерживает выход из системы по обратному каналу.
Формат политики NameID	Ссылка URI, соответствующая формату идентификатора имени. По умолчанию Tuxedo SSO устанавливает его в `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`.
Основной тип	Указывает, какая часть утверждения SAML будет использоваться для идентификации и отслеживания внешних идентификаторов пользователей. Может быть либо Subject NameID, либо атрибутом SAML (по имени или по понятному имени). Значение Subject NameID не может быть установлено вместе со значением формата политики NameID 'urn:oasis:names:tc:SAML:2.0:nameid-format:transient'.
Основной атрибут	Если тип Principal не является пустым, в этом поле указывается имя («Атрибут [Имя]») или понятное имя («Атрибут [Дружественное имя]») идентифицирующего атрибута.
Разрешить создание	Разрешить внешнему поставщику удостоверений создать новый идентификатор, представляющий принципала.
Ответ на привязку HTTP-POST	Управляет привязкой SAML в ответ на любые запросы SAML, отправленные внешним IDP. Когда **OFF** , Tuxedo SSO использует Redirect Binding.
Ответ на привязку ARTIFACT	Управляет привязкой SAML в ответ на любые запросы SAML, отправленные внешним IDP. Когда **OFF** , Tuxedo SSO оценивает конфигурацию ответа привязки HTTP-POST.
Привязка HTTP-POST для AuthnRequest	Управляет привязкой SAML при запросе аутентификации от внешнего IDP. Когда **OFF** , Tuxedo SSO использует Redirect Binding.
Хочу подписанные AuthnRequests	Если **установлено значение ON** , Tuxedo SSO использует пару ключей области для подписи запросов, отправляемых внешнему SAML IDP.
Хочу Подписанные Утверждения	Указывает, ожидает ли данный поставщик услуг подписанное утверждение.
Хотите	Указывает, ожидает ли данный поставщик услуг зашифрованное

Конфигурация	Описание
зашифровать утверждения	утверждение.
Алгоритм подписи	Если **Want AuthnRequests Signed** имеет значение **ON** , алгоритм подписи для использования. Обратите внимание, что `SHA1`основанные алгоритмы устарели и могут быть удалены в будущем выпуске. Мы рекомендуем использовать более безопасный алгоритм вместо `*_SHA1`. Кроме того, с `*_SHA1`алгоритмами проверка подписей не работает, если поставщик удостоверений SAML (например, другой экземпляр Tuxedo SSO) работает на Java 17 или выше.
Алгоритм шифрования	Алгоритм шифрования, который используется SAML IDP для шифрования документов SAML, утверждений или идентификаторов. Соответствующий ключ дешифрования для дешифрования частей документа SAML будет выбран на основе этого настроенного алгоритма и должен быть доступен в ключах области для использования шифрования (ENC). Если алгоритм не настроен, разрешен любой поддерживаемый алгоритм, и ключ дешифрования будет выбран на основе алгоритма, указанного в самом документе SAML.
Имя ключа подписи SAML	Подписанные документы SAML, отправленные с использованием привязки POST, содержат идентификацию ключа подписи в `KeyName`элементе, который по умолчанию содержит идентификатор ключа Tuxedo SSO. Внешние поставщики идентификации SAML могут ожидать другое имя ключа. Этот переключатель определяет, `KeyName`содержит ли: * `KEY_ID`- идентификатор ключа. * `CERT_SUBJECT`- субъект из сертификата, соответствующий ключу области. Службы федерации Microsoft Active Directory ожидают `CERT_SUBJECT`. * `NONE`- Tuxedo SSO опускает подсказку имени ключа из сообщения SAML.
Принудительная аутентификация	Пользователь должен ввести свои учетные данные на внешнем IDP, даже если он уже вошел в систему.
Проверить подпись	Если **установлено значение ON** , область ожидает, что запросы и ответы SAML от внешнего IDP будут иметь цифровую подпись.
URL-адрес дескриптора метаданных	Внешний URL, где поставщик удостоверений публикует `IDPSSODescriptor`метаданные. Этот URL используется для загрузки сертификатов поставщика удостоверений при нажатии на `Reload keys`или действия.`Import keys`

Конфигурация	Описание
Использовать URL-адрес дескриптора метаданных	Если **включено** , сертификаты для проверки подписей автоматически загружаются из `Metadata descriptor URL`и кэшируются в Tuxedo SSO. Поставщик SAML может проверять подписи двумя разными способами. Если запрашивается определенный сертификат (обычно в `POST`привязке), а его нет в кэше, сертификаты автоматически обновляются с URL. Если запрашиваются все сертификаты для проверки подписи (`REDIRECT`привязка), обновление выполняется только по истечении максимального времени кэширования (см. public-key-storage spi в руководстве по конфигурации всех поставщиков для получения дополнительной информации о том, как работает кэш). Кэш также можно обновить вручную с помощью действия `Reload Keys`на странице поставщика удостоверений. Если опция **выключена** , сертификаты `Validating X509 Certificates`используются для проверки подписей.
Проверка сертификатов X509	Публичные сертификаты, которые Tuxedo SSO использует для проверки подписей запросов и ответов SAML от внешнего IDP, когда **выключено**`Use metadata descriptor URL` . Можно ввести несколько сертификатов, разделенных запятой ( ). Сертификаты можно повторно импортировать, щелкнув действие на странице поставщика удостоверений. Действие загружает текущие сертификаты в конечную точку метаданных и назначает их конфигурации в этой же опции. Вам нужно щелкнуть , чтобы определенно сохранить повторно импортированные сертификаты.`,Metadata descriptor URLImport KeysSave`
Метаданные поставщика услуг подписи	Если **включено** , Tuxedo SSO использует пару ключей области для подписи дескриптора метаданных поставщика услуг SAML .
Пройти предмет	Контролирует, пересылает ли Tuxedo SSO `login_hint`параметр запроса в IDP. Tuxedo SSO добавляет значение этого поля в параметр login_hint в теме AuthnRequest, чтобы поставщики назначения могли предварительно заполнить свою форму входа.
Индекс потребления атрибутов	Определяет набор атрибутов для запроса к удаленному IDP. Tuxedo SSO автоматически добавляет атрибуты, сопоставленные в конфигурации поставщика удостоверений, в автоматически сгенерированный документ метаданных SP.
Имя службы	Описательное имя для набора атрибутов, которые объявляются в

Конфигурация	Описание
потребления атрибутов	автоматически сгенерированном документе метаданных SP.

Вы можете импортировать все данные конфигурации, указав URL или файл, указывающий на дескриптор сущности SAML IDP внешнего IDP. Если вы подключаетесь к внешнему IDP Tuxedo SSO, вы можете импортировать настройки IDP из URL <root>/realms/{realm-name}/protocol/saml/descriptor. Эта ссылка представляет собой XML-документ, описывающий метаданные об IDP. Вы также можете импортировать все эти данные конфигурации, указав URL или XML-файл, указывающий на дескриптор сущности внешнего SAML IDP для подключения.

Запрос определенных AuthnContexts

Поставщики удостоверений облегчают клиентам задание ограничений на метод аутентификации, проверяющий личность пользователя. Например, запрос MFA, аутентификации Kerberos или требований безопасности. Эти ограничения используют определенные критерии AuthnContext. Клиент может запросить один или несколько критериев и указать, как поставщик удостоверений должен соответствовать запрошенному AuthnContext, точно или путем удовлетворения других эквивалентов.

Вы можете перечислить критерии, требуемые вашим поставщиком услуг, добавив ClassRefs или DeclRefs в раздел Requested AuthnContext Constraints. Обычно вам нужно предоставить либо ClassRefs, либо DeclRefs, поэтому проверьте в документации вашего поставщика удостоверений, какие значения поддерживаются. Если ClassRefs или DeclRefs отсутствуют, поставщик удостоверений не применяет дополнительные ограничения.

Таблица 4. Запрошенные ограничения AuthnContext

Конфигурация	Описание
Сравнение	Метод, который поставщик удостоверений использует для оценки требований контекста. Доступные значения: `Exact`, `Minimum`, `Maximum`, или `Better`. Значение по умолчанию: `Exact`.
Классы AuthnContextRefs	AuthnContext ClassRefs, описывающий требуемые критерии.

Конфигурация	Описание
AuthnContext DeclRefs	AuthnContext DeclRefs, описывающий требуемые критерии.

SP-дескриптор

При доступе к метаданным SAML SP провайдера найдите Endpointsэлемент в настройках конфигурации провайдера идентификации. Он содержит SAML 2.0 Service Provider Metadataссылку, которая генерирует дескриптор сущности SAML для провайдера услуг. Вы можете загрузить дескриптор или скопировать его URL, а затем импортировать его в удаленный провайдер идентификации.

Эти метаданные также доступны публично, перейдя по следующему URL-адресу:

http[s]://{host:port}/realms/{realm-name}/broker/{broker-alias}/endpoint/descriptor

Перед доступом к дескриптору обязательно сохраните все изменения конфигурации.

Отправить тему в запросах SAML

По умолчанию социальная кнопка, указывающая на поставщика удостоверений SAML, перенаправляет пользователя на следующий URL-адрес для входа:

http[s]://{host:port}/realms/${realm-name}/broker/{broker-alias}/login

Добавление параметра запроса с именем login_hintк этому URL добавляет значение параметра к запросу SAML как атрибут Subject. Если этот параметр запроса пуст, Tuxedo SSO не добавляет тему к запросу.

Включите опцию «Передавать тему», чтобы отправлять тему в запросах SAML.

Поставщик удостоверений, предложенный клиентом

Приложения OIDC могут обходить страницу входа Tuxedo SSO, намекая на поставщика удостоверений, которого они хотят использовать. Вы можете включить это, установив kc_idp_hintпараметр запроса в конечной точке авторизации Authorization Code Flow.

С помощью клиентских адаптеров Tuxedo SSO OIDC вы можете указать этот параметр запроса при доступе к защищенному ресурсу в приложении.

Например:

```
GET /myapplication.com?kc_idp_hint=facebook HTTP/1.1
Host: localhost:8080
```

В этом случае в вашей области должен быть поставщик идентификации с facebookпсевдонимом. Если этот поставщик не существует, отображается форма входа.

Если вы используете адаптер JavaScript, вы можете добиться того же поведения следующим образом:

```
const Tuxedo SSO = new Tuxedo SSO({
        url: "http://Tuxedo SSO-server",
        realm: "my-realm",
        clientId: "my-app"
});

await Tuxedo SSO.createLoginUrl({
        idpHint: 'facebook'
});
```

С kc_idp_hintпараметром запроса клиент может переопределить поставщика идентификации по умолчанию, если вы настроите его для Identity Provider Redirectorаутентификатора. Клиент может отключить автоматическое перенаправление, установив kc_idp_hintпараметр запроса на пустое значение.

Картографирование претензий и утверждений

Вы можете импортировать метаданные SAML и OpenID Connect, предоставленные внешним IDP, с которым вы проходите аутентификацию, в область. После импорта вы можете извлечь метаданные профиля пользователя и другую информацию, чтобы сделать ее доступной для ваших приложений.

Каждый пользователь, входящий в вашу область с помощью внешнего поставщика удостоверений, имеет запись в локальной базе данных Tuxedo SSO, основанную на метаданных из утверждений и заявлений SAML или OIDC.

Процедура

1. В меню выберите «Поставщики удостоверений» .
2. Выберите одного из поставщиков удостоверений в списке.
3. Нажмите на вкладку «Картографы» .

   Картографы поставщиков удостоверений
4. Нажмите Добавить картографа .

   Сопоставитель поставщиков удостоверений
5. Выберите значение для Sync Mode Override . Mapper обновляет информацию о пользователе, когда пользователи повторно входят в систему в соответствии с этой настройкой.
   a. Выберите Legacy , чтобы использовать поведение предыдущей версии Tuxedo SSO.
   b. Выберите импорт , чтобы импортировать данные с момента, когда пользователь был впервые создан в Tuxedo SSO во время первого входа в Tuxedo SSO с использованием определенного поставщика удостоверений.
   c. Выберите принудительное обновление пользовательских данных при каждом входе пользователя в систему.
   d. Выберите inherit , чтобы использовать режим синхронизации, настроенный в поставщике удостоверений. Все остальные параметры переопределят этот режим синхронизации.
6. Выберите картографа из списка Тип картографа . Наведите курсор на Тип картографа, чтобы увидеть описание картографа и конфигурацию для ввода для картографа.
7. Нажмите «Сохранить ».

Для утверждений на основе JSON можно использовать точечную нотацию для вложенности и квадратные скобки для доступа к полям массива по индексу. Например, contact.address[0].country.

Чтобы исследовать структуру данных JSON профиля пользователя, предоставляемых социальными провайдерами, вы можете включить DEBUGрегистратор уровня org.Tuxedo SSO.social.user_profile_dumpпри запуске сервера.

Доступные данные сеанса пользователя

После входа пользователя из внешнего IDP Tuxedo SSO сохраняет данные заметок сеанса пользователя, к которым вы можете получить доступ. Эти данные могут быть распространены на клиента, запрашивающего вход, с использованием токена или утверждения SAML, переданного обратно клиенту с использованием соответствующего клиентского картографа.

поставщик_идентификации

> Псевдоним IDP брокера, используемый для выполнения входа.

идентификатор_поставщика_идентификации

> Имя пользователя IDP текущего аутентифицированного пользователя. Часто, но не всегда, совпадает с именем пользователя Tuxedo SSO. Например, Tuxedo SSO может связать пользователя john` с пользователем Facebook john123@gmail.com. В этом случае значение примечания сеанса пользователя равно john123@gmail.com.

Вы можете использовать Protocol MapperUser Session Note для распространения этой информации среди своих клиентов.

Первый процесс входа в систему

Когда пользователи входят в систему через брокерскую идентификацию, Tuxedo SSO импортирует и связывает аспекты пользователя в локальной базе данных

области. Когда Tuxedo SSO успешно аутентифицирует пользователей через внешнего поставщика идентификации, могут существовать две ситуации:

- Tuxedo SSO уже импортировал и связал учетную запись пользователя с учетной записью аутентифицированного поставщика удостоверений. В этом случае Tuxedo SSO аутентифицируется как существующий пользователь и перенаправляет обратно в приложение.
- Для этого пользователя в Tuxedo SSO не существует учетной записи. Обычно вы регистрируетесь и импортируете новую учетную запись в базу данных Tuxedo SSO, но может быть существующая учетная запись Tuxedo SSO с тем же адресом электронной почты. Автоматическое связывание существующей локальной учетной записи с внешним поставщиком удостоверений является потенциальной дырой в безопасности. Вы не всегда можете доверять информации, которую получаете от внешнего поставщика удостоверений.

Разные организации предъявляют разные требования к некоторым из этих ситуаций. С Tuxedo SSO вы можете использовать First Login Flowопцию в настройках IDP, чтобы выбрать рабочий процесс для пользователя, впервые входящего в систему с внешнего IDP. По умолчанию First Login Flowопция указывает на first broker loginпоток, но вы можете использовать свой поток или другие потоки для разных поставщиков удостоверений.

Поток находится в консоли администратора на вкладке Аутентификация . При выборе First Broker Loginпотока вы видите аутентификаторы, используемые по умолчанию. Вы можете перенастроить существующий поток. Например, вы можете отключить некоторые аутентификаторы, пометить некоторые из них как requiredили настроить некоторые аутентификаторы.

Вы также можете создать новый поток аутентификации, написать собственные реализации Authenticator и использовать его в своем потоке. Для получения дополнительной информации см. Руководство разработчика сервера .

Аутентификаторы первого входа по умолчанию

Обзор профиля

- Этот аутентификатор отображает страницу с информацией о профиле, чтобы пользователи могли просматривать свой профиль, который Tuxedo SSO получает от поставщика удостоверений.

- Эту опцию можно установить Update Profile On First Loginв меню «Действия» .

- Если установлено значение ON , пользователям отображается страница профиля, запрашивающая дополнительную информацию для объединения идентификационных данных пользователя.

- При отсутствии этого параметра пользователям отображается страница профиля, если поставщик удостоверений не предоставляет обязательную информацию, такую как адрес электронной почты, имя или фамилия.

- Если установлено значение OFF , страница профиля не отображается, пока пользователь позднее не нажмет на Review profile infoссылку на странице, отображаемой аутентификатором Confirm Link Existing Account.

Создать пользователя, если он уникален

Этот аутентификатор проверяет, существует ли уже существующая учетная запись Tuxedo SSO с тем же адресом электронной почты или именем пользователя, что и у учетной записи поставщика удостоверений. Если нет, то аутентификатор просто создает новую локальную учетную запись Tuxedo SSO и связывает ее с поставщиком удостоверений, и весь поток завершается. В противном случае он переходит к следующему Handle Existing Accountподпотоку. Если вы всегда хотите убедиться, что нет дублирующейся учетной записи, вы можете пометить этот аутентификатор как REQUIRED. В этом случае пользователь увидит страницу с ошибкой, если существует существующая учетная запись Tuxedo SSO, и пользователю нужно будет связать учетную запись поставщика удостоверений через Управление учетными записями.

- Этот аутентификатор проверяет, существует ли учетная запись Tuxedo SSO с тем же адресом электронной почты или именем пользователя, что и у учетной записи поставщика удостоверений.

- Если учетная запись не существует, аутентификатор создает локальную учетную запись Tuxedo SSO, связывает ее с поставщиком удостоверений и завершает процесс.

- Если учетная запись существует, аутентификатор реализует следующий Handle Existing Accountподпоток.

- Чтобы убедиться в отсутствии дублирующейся учетной записи, вы можете пометить этот аутентификатор как REQUIRED. Пользователь видит страницу с ошибкой, если учетная запись Tuxedo SSO существует, и пользователи должны связать свою учетную запись поставщика удостоверений через Управление учетными записями.

Подтвердите привязку существующей учетной записи

- На странице информации пользователи видят учетную запись Tuxedo SSO с тем же адресом электронной почты. Пользователи могут снова просмотреть свой профиль и использовать другой адрес электронной почты или имя пользователя. Поток перезапускается и возвращается к Review Profileаутентификатору.

- Кроме того, пользователи могут подтвердить, что они хотят связать свою учетную запись поставщика удостоверений с существующей учетной записью Tuxedo SSO.

- Отключите этот аутентификатор, если вы не хотите, чтобы пользователи видели эту страницу подтверждения, и переходите сразу к привязке учетной записи поставщика удостоверений с помощью проверки по электронной почте или повторной аутентификации.

Подтвердите существующую учетную запись по электронной почте

- Этот аутентификатор используется ALTERNATIVEпо умолчанию. Tuxedo SSO использует этот аутентификатор, если в области настроена настройка SMTP.

- Аутентификатор отправляет пользователям электронное письмо с подтверждением того, что они хотят связать поставщика удостоверений со своей учетной записью Tuxedo SSO.

- Отключите этот аутентификатор, если вы не хотите подтверждать привязку по электронной почте, но хотите, чтобы пользователи повторно проходили аутентификацию с помощью своего пароля.

Подтвердите существующую учетную запись путем повторной аутентификации

- Используйте этот аутентификатор, если аутентификатор электронной почты недоступен. Например, вы не настроили SMTP для своей области. Этот аутентификатор отображает экран входа для аутентификации пользователей, чтобы связать их учетную запись Tuxedo SSO с поставщиком удостоверений.

- Пользователи также могут повторно пройти аутентификацию с помощью другого поставщика удостоверений, уже привязанного к их учетной записи Tuxedo SSO.

- Вы можете заставить пользователей использовать OTP. В противном случае это необязательно и используется, если вы установили OTP для учетной записи пользователя.

Автоматически связать существующий первый поток входа в систему

> Аутентификатор AutoLink опасен в общей среде, где пользователи могут регистрироваться, используя произвольные имена пользователей или адреса электронной почты. Не используйте этот аутентификатор, если вы не тщательно курируете регистрацию пользователей и не назначаете имена пользователей и адреса электронной почты.

Чтобы настроить первый поток входа, который автоматически связывает пользователей без запроса, создайте новый поток со следующими двумя аутентификаторами:

Создать пользователя, если он уникален

> Этот аутентификатор гарантирует, что Tuxedo SSO обрабатывает уникальных пользователей. Установите требование аутентификатора на Альтернативный .

Автоматически установить существующего пользователя

> Этот аутентификатор устанавливает существующего пользователя в контекст аутентификации без проверки. Установите требование аутентификатора на «Альтернативный».

Эта настройка является самой простой из доступных, но можно использовать и другие аутентификаторы. Например: * Вы можете добавить аутентификатор Review Profile в начало потока, если хотите, чтобы конечные пользователи подтверждали информацию своего профиля. * Вы можете добавить механизмы аутентификации в этот поток, заставив пользователя подтвердить свои учетные данные. Добавление механизмов аутентификации требует сложного потока. Например, вы можете установить «Автоматически установить существующего пользователя» и «Форму пароля» как «Обязательные» в подпотоке «Альтернативный».

Отключение автоматического создания пользователей

Поток первого входа по умолчанию ищет учетную запись Tuxedo SSO, соответствующую внешнему идентификатору, и предлагает связать их. Если соответствующей учетной записи Tuxedo SSO не существует, поток автоматически создает ее.

Это поведение по умолчанию может быть неподходящим для некоторых настроек. Одним из примеров является использование хранилища пользователей LDAP только для чтения, где все пользователи предварительно созданы. В этом случае необходимо отключить автоматическое создание пользователей.

Чтобы отключить создание пользователей:

Процедура

1. Нажмите «Аутентификация» в меню.
2. Выберите из списка «Первый брокерский логин» .
3. Установите для параметра «Создать уникального пользователя» значение «ОТКЛЮЧЕНО » .
4. Установите для параметра Подтверждение привязки существующей учетной записи значение ОТКЛЮЧЕНО .

Эта конфигурация также подразумевает, что сам Tuxedo SSO не сможет определить, какой внутренний аккаунт будет соответствовать внешней личности. Поэтому Verify Existing Account By Re-authenticationаутентификатор попросит пользователя предоставить как имя пользователя, так и пароль.

Включение или выключение создания пользователя поставщиком удостоверений полностью не зависит от переключателя регистрации пользователя области . Вы можете включить создание

пользователя поставщиком удостоверений и в то же время отключить самостоятельную регистрацию пользователя в настройках входа в область или наоборот.

Определить существующий процесс первого входа пользователя

Чтобы настроить первый процесс входа в систему, в котором:

- войти в систему могут только пользователи, уже зарегистрированные в этой области,
- пользователи автоматически подключаются без предварительного запроса,

создайте новый поток со следующими двумя аутентификаторами:

Определить существующего пользователя-брокера

> Этот аутентификатор обеспечивает обработку уникальных пользователей. Установите требование аутентификатора на REQUIRED.

Автоматически установить существующего пользователя

> Автоматически устанавливает существующего пользователя в контекст аутентификации без какой-либо проверки. Установите требование аутентификатора на REQUIRED.

Вам необходимо установить First Login Flowконфигурацию поставщика удостоверений для этого потока. Вы также можете установить set Sync Modeдля, forceесли вы хотите обновить профиль пользователя (фамилия, имя…) с атрибутами поставщика удостоверений.

> Этот поток можно использовать, если вы хотите делегировать идентификационные данные другим поставщикам идентификационных данных (например, GitHub, Facebook…), но хотите управлять тем, какие пользователи могут входить в систему.

При такой конфигурации Tuxedo SSO не может определить, какой внутренний аккаунт соответствует внешнему идентификатору. Аутентификатор Verify Existing Account By Re-authentication запрашивает у провайдера имя пользователя и пароль.

Переопределить существующую ссылку брокера

Если необходимо связать другую учетную запись с той же учетной записью Tuxedo SSO в рамках того же поставщика удостоверений, вы можете настроить следующий аутентификатор.

Подтвердите переопределение существующей ссылки

> Этот аутентификатор обнаружит существующую ссылку брокера для пользователя и отобразит страницу подтверждения для подтверждения переопределения существующей ссылки брокера. Установите требование аутентификатора на ОБЯЗАТЕЛЬНО.

Типичным применением этого аутентификатора является следующий сценарий:

- Например, рассмотрим пользователя Tuxedo SSO johnc адресом электронной почты john@gmail.com. Этот пользователь связан с поставщиком удостоверений googlec googleименем пользователя john@gmail.com.
- Затем, например, пользователь Tuxedo SSO johnсоздает новую учетную запись Google с адресом электронной почты.john-new@gmail.com
- Затем во время входа в Tuxedo SSO пользователь проходит аутентификацию у поставщика удостоверений googlec новым именем пользователя, например john-new@gmail.com, , которое еще не связано ни с одной учетной записью Tuxedo SSO (поскольку учетная запись Tuxedo SSO johnвсе еще связана с googlepользователем john@gmail.com), и, следовательно, запускается процесс входа в систему через первого брокера.
- Во время входа в систему первого брокера пользователь Tuxedo SSO johnкаким-то образом аутентифицируется (либо путем повторной аутентификации при входе в систему первого брокера по умолчанию, либо, например, с помощью аутентификатора, например Detect existing broker user).
- Теперь, используя этот аутентификатор в потоке аутентификации, можно переопределить ссылку IDP на googlepпоставщика удостоверений пользователя Tuxedo SSO johnновой googlessылкой на googlepпользователя john-new@gmail.comпосле того, как пользователь johnподтвердит ее.

При создании потоков аутентификации с помощью этого аутентификатора обязательно добавьте этот аутентификатор после того, как другие аутентификаторы уже установлены пользователем Tuxedo SSO другими способами

(путем повторной аутентификации или после того, Detect existing broker userкак указано выше).

Извлечение внешних токенов IDP

С помощью Tuxedo SSO вы можете хранить токены и ответы процесса аутентификации с помощью внешнего IDP, используя Store Tokenпараметр конфигурации на странице настроек IDP.

Код приложения может извлекать эти токены и ответы для импорта дополнительной информации пользователя или для безопасного запроса внешнего IDP. Например, приложение может использовать токен Google для использования других служб Google и REST API. Чтобы извлечь токен для определенного поставщика удостоверений, отправьте запрос следующим образом:

```
GET /realms/{realm}/broker/{provider_alias}/token HTTP/1.1
Host: localhost:8080
Authorization: Bearer <Tuxedo SSO ACCESS TOKEN>
```

Приложение должно пройти аутентификацию в Tuxedo SSO и получить токен доступа. Этот токен доступа должен иметь установленную brokerроль на уровне клиента read-token, поэтому у пользователя должно быть сопоставление ролей для этой роли, а клиентское приложение должно иметь эту роль в своей области действия. В этом случае, поскольку вы получаете доступ к защищенной службе в Tuxedo SSO, отправьте токен доступа, выданный Tuxedo SSO во время аутентификации пользователя. Вы можете назначить эту роль недавно импортированным пользователям на странице конфигурации брокера, установив переключатель Stored Tokens Readable в положение ON .

Эти внешние токены можно восстановить, повторно войдя в систему через провайдера или используя API привязки учетных записей, инициированный клиентом.

Выход из брокера идентификации

При выходе из системы Tuxedo SSO отправляет запрос внешнему поставщику удостоверений, который использовался для первоначального входа в систему, и выполняет выход пользователя из этого поставщика удостоверений.

протоколы единого входа

В этом разделе рассматриваются протоколы аутентификации, сервер аутентификации Tuxedo SSO и то, как приложения, защищенные сервером аутентификации Tuxedo SSO, взаимодействуют с этими протоколами.

OpenID-подключение

OpenID Connect (OIDC) — это протокол аутентификации, являющийся расширением OAuth 2.0 .

OAuth 2.0 — это фреймворк для построения протоколов авторизации, и он неполный. Однако OIDC — это полноценный протокол аутентификации и авторизации, использующий стандарты Json Web Token (JWT). Стандарты JWT определяют формат JSON токена идентификации и методы цифровой подписи и шифрования данных компактным и удобным для веб-доступа способом.

В целом, OIDC реализует два варианта использования. Первый случай — это приложение, запрашивающее, чтобы сервер Tuxedo SSO аутентифицировал пользователя. После успешного входа в систему приложение получает токен идентификации и токен доступа . Токен идентификации содержит информацию о пользователе, включая имя пользователя, адрес электронной почты и информацию о профиле. Область цифровой подписью подписывает токен доступа , содержащий информацию о доступе (например, сопоставления ролей пользователей), которую приложения используют для определения ресурсов, к которым пользователи могут получить доступ в приложении.

Второй вариант использования — клиент, получающий доступ к удаленным сервисам.

- Клиент запрашивает токен доступа у Tuxedo SSO для вызова удаленных служб от имени пользователя.
- Tuxedo SSO аутентифицирует пользователя и запрашивает у него согласие на предоставление доступа запрашивающему клиенту.
- Клиент получает токен доступа , подписанный цифровой подписью области.
- Клиент отправляет REST-запросы на удаленные сервисы, используя токен доступа .
- Удаленная служба REST извлекает токен доступа .
- Удаленная служба REST проверяет подпись токенов.
- Удаленная служба REST принимает решение, основываясь на информации о доступе в токене, обработать или отклонить запрос.

Потоки аутентификации OIDC

OIDC имеет несколько методов или потоков, которые клиенты или приложения могут использовать для аутентификации пользователей и получения токенов идентификации и доступа . Метод зависит от типа приложения или клиента, запрашивающего доступ.

Поток кода авторизации
Authorization Code Flow — это протокол на основе браузера, который подходит для аутентификации и авторизации приложений на основе браузера. Он использует перенаправления браузера для получения идентификаторов и токенов доступа .

1. Пользователь подключается к приложению с помощью браузера. Приложение обнаруживает, что пользователь не вошел в приложение.
2. Приложение перенаправляет браузер на Tuxedo SSO для аутентификации.
3. Приложение передает URL обратного вызова как параметр запроса в перенаправлении браузера. Tuxedo SSO использует параметр при успешной аутентификации.

4. Tuxedo SSO аутентифицирует пользователя и создает одноразовый, кратковременный, временный код.
5. Tuxedo SSO перенаправляет в приложение, используя URL-адрес обратного вызова, и добавляет временный код в качестве параметра запроса в URL-адрес обратного вызова.
6. Приложение извлекает временный код и выполняет фоновый вызов REST в Tuxedo SSO для обмена кода на идентификатор и доступ и токен обновления . Для предотвращения атак повторного воспроизведения временный код нельзя использовать более одного раза.

Система уязвима для украденного токена на протяжении всего срока действия этого токена. По соображениям безопасности и масштабируемости токены доступа обычно устанавливаются на быстрый срок действия, поэтому последующие запросы токенов не будут иметь успеха. Если токен истекает, приложение может получить новый токен доступа, используя дополнительный токен *обновления*, отправленный протоколом входа.

Конфиденциальные клиенты предоставляют клиентские секреты при обмене временных кодов на токены. *Публичные* клиенты не обязаны предоставлять клиентские секреты. *Публичные* клиенты защищены, когда HTTPS строго соблюдается, а URI перенаправления, зарегистрированные для клиента, строго контролируются. Клиенты HTML5/JavaScript должны быть *публичными* клиентами, поскольку нет способа безопасно передать клиентский секрет клиентам HTML5/JavaScript. Более подробную информацию см. в главе Управление клиентами .

Tuxedo SSO также поддерживает спецификацию Proof Key for Code Exchange .

Неявный поток

Implicit Flow — это протокол на основе браузера. Он похож на Authorization Code Flow, но с меньшим количеством запросов и без токенов обновления.

Существует вероятность утечки токенов *доступа* в историю браузера при передаче токенов через URI перенаправления (см. ниже).

Кроме того, этот поток не предоставляет клиентам токены обновления. Поэтому токены доступа должны быть долгосрочными, или пользователи должны повторно проходить аутентификацию по истечении срока их действия.

Мы не советуем использовать этот поток. Этот поток поддерживается, поскольку он есть в

спецификации OIDC и OAuth 2.0.

Протокол работает следующим образом:

1. Пользователь подключается к приложению с помощью браузера. Приложение обнаруживает, что пользователь не вошел в приложение.
2. Приложение перенаправляет браузер на Tuxedo SSO для аутентификации.
3. Приложение передает URL обратного вызова как параметр запроса в перенаправлении браузера. Tuxedo SSO использует параметр запроса при успешной аутентификации.
4. Tuxedo SSO аутентифицирует пользователя и создает идентификатор и токен доступа . Tuxedo SSO перенаправляет в приложение с помощью URL обратного вызова и дополнительно добавляет идентификатор и токены доступа в качестве параметра запроса в URL обратного вызова.
5. Приложение извлекает идентификационные данные и токены доступа из URL-адреса обратного вызова.

Предоставление учетных данных пароля владельца ресурса (предоставление прямого доступа)
Direct Access Grants используются клиентами REST для получения токенов от имени пользователей. Это HTTP POST-запрос, который содержит:

- Учетные данные пользователя. Учетные данные отправляются в параметрах формы.
- Идентификатор клиента.
- Секрет клиента (если это конфиденциальный клиент).

HTTP-ответ содержит токены идентификации , доступа и обновления .

Предоставление клиентских учетных данных
Client Credentials Grant создает токен на основе метаданных и разрешений учетной записи службы, связанной с клиентом, вместо получения токена, который работает от имени внешнего пользователя. Client Credentials Grant используются клиентами REST.

Более подробную информацию см. в главе «Учетные записи служб» .

Грант обновления токена

По умолчанию Tuxedo SSO возвращает токены обновления в ответах токенов от большинства потоков. Некоторые исключения — неявный поток или предоставление клиентских учетных данных, описанные выше.

Refresh token привязан к сеансу пользователя сеанса браузера SSO и может быть действительным в течение всего срока действия сеанса пользователя. Однако этот клиент должен отправлять запрос refresh-token по крайней мере один раз за указанный интервал. В противном случае сеанс может считаться "простаивающим" и может истечь. Для получения дополнительной информации см. раздел тайм-ауты .

Tuxedo SSO поддерживает автономные токены , которые обычно можно использовать, когда клиенту необходимо использовать токен обновления, даже если соответствующий сеанс единого входа браузера уже истек.

Обновить ротацию токенов

Можно указать, что токен обновления считается недействительным после его использования. Это означает, что клиент должен всегда сохранять токен обновления из последнего ответа обновления, поскольку старые токены обновления, которые уже использовались, больше не будут считаться действительными Tuxedo SSO. Это можно установить с помощью параметра Revoke Refresh token , как указано в разделе тайм-аутов .

Tuxedo SSO также поддерживает ситуацию, когда не существует ротации токенов обновления. В этом случае токен обновления возвращается во время входа в систему, но последующие ответы на запросы токенов обновления не будут возвращать новые токены обновления. Такая практика рекомендуется, например, в черновой спецификации FAPI 2 в разделе «Безопасность приложений» . В Tuxedo SSO можно пропустить ротацию токенов обновления с помощью политик клиента . Вы можете добавить исполнителя suppress-refresh-token-rotationв какой-либо профиль клиента и настроить политику клиента, чтобы указать, для каких клиентов будет запущен профиль, что означает, что для этих клиентов ротация токенов обновления будет пропущена.

Предоставление разрешения на использование устройства
Это используется клиентами, работающими на подключенных к Интернету устройствах, которые имеют ограниченные возможности ввода или не имеют подходящего браузера. Вот краткое описание протокола:

1. Приложение запрашивает у Tuxedo SSO код устройства и код пользователя. Tuxedo SSO создает код устройства и код пользователя. Tuxedo SSO возвращает ответ, включающий код устройства и код пользователя, приложению.
2. Приложение предоставляет пользователю код пользователя и URI проверки. Пользователь получает доступ к URI проверки для аутентификации с помощью другого браузера. Вы можете определить короткий verification_uri, который будет перенаправлен на URI проверки Tuxedo SSO (/realms/realm_name/device) за пределами Tuxedo SSO, например, в прокси.
3. Приложение многократно опрашивает Tuxedo SSO, чтобы узнать, завершил ли пользователь авторизацию. Если аутентификация пользователя завершена, приложение обменивает код устройства на идентификатор , доступ и токен обновления .

Клиент инициировал предоставление аутентификации обратного канала
Эта функция используется клиентами, которые хотят инициировать поток аутентификации, напрямую связываясь с поставщиком OpenID без перенаправления через браузер пользователя, как при предоставлении кода авторизации OAuth 2.0. Вот краткое описание протокола:

1. Клиент запрашивает у Tuxedo SSO auth_req_id, который идентифицирует запрос аутентификации, сделанный клиентом. Tuxedo SSO создает auth_req_id.
2. После получения auth_req_id этому клиенту необходимо неоднократно опрашивать Tuxedo SSO, чтобы получить токен доступа, токен обновления и токен идентификатора от Tuxedo SSO в обмен на auth_req_id, пока пользователь не будет аутентифицирован.

Администратор может настроить операции, связанные с проверкой подлинности клиентского канала (CIBA), в CIBA Policyсоответствии с областью.

Также, пожалуйста, ознакомьтесь с другими разделами документации Tuxedo SSO, такими как Backchannel Authentication Endpoint и Client Initiated Backchannel Authentication Grant в разделе «Защита приложений» .

Политика CIBA

Администратор выполняет следующие операции на Admin Console:

- Откройте Authentication → CIBA Policyвкладку.
- Настройте элементы и нажмите Save.

Далее следуют настраиваемые элементы и их описание.

Конфигурация	Описание
Режим доставки токена обратного канала	Указание того, как CD (Consumption Device) получает результат аутентификации и соответствующие токены. Существует три режима: «poll», «ping» и «push». Tuxedo SSO поддерживает только «poll». Настройка по умолчанию — «poll». Эта конфигурация обязательна. Для получения более подробной информации см. Спецификацию CIBA .
Истекает через	Время истечения срока действия "auth_req_id" в секундах с момента получения запроса на аутентификацию. Значение по умолчанию — 120. Эта конфигурация обязательна. Для получения более подробной информации см. Спецификацию CIBA .
Интервал	Интервал в секундах, который CD (Consumption Device) должен ждать между запросами опроса к конечной точке токена. Значение по умолчанию — 5. Эта конфигурация необязательна. Для получения более подробной информации см. Спецификацию CIBA .
Подсказка пользователя, запрошенная аутентификация	Способ идентификации конечного пользователя, для которого запрашивается аутентификация. Значение по умолчанию — «login_hint». Существует три режима: «login_hint», «login_hint_token» и «id_token_hint». Tuxedo SSO поддерживает только «login_hint». Эта конфигурация обязательна. Для получения более подробной информации см. Спецификацию CIBA .

Настройка провайдера

Грант CIBA использует следующих двух поставщиков.

1. Поставщик канала аутентификации: обеспечивает связь между Tuxedo SSO и объектом, который фактически аутентифицирует пользователя через AD (устройство аутентификации).
2. Поставщик распознавания пользователей: получает UserModelот Tuxedo SSO информацию, предоставленную клиентом, для идентификации пользователя.

Tuxedo SSO имеет обоих поставщиков по умолчанию. Однако администратору необходимо настроить поставщика канала аутентификации следующим образом:

kc.[sh|bat] start --spi-ciba-auth-channel-ciba-http-auth-channel-http-authentication-channel-uri=https://backend.internal.example.com

Далее следуют настраиваемые элементы и их описание.

Конфигурация	Описание
http-аутентификация-канал-uri	Указание URI сущности, которая фактически аутентифицирует пользователя через AD (устройство аутентификации).

Поставщик канала аутентификации
Стандартный документ CIBA не определяет, как аутентифицировать пользователя с помощью AD. Поэтому это может быть реализовано по усмотрению продуктов. Tuxedo SSO делегирует эту аутентификацию внешнему объекту аутентификации. Для связи с объектом аутентификации Tuxedo SSO предоставляет поставщика канала аутентификации.

Его реализация Tuxedo SSO предполагает, что сущность аутентификации находится под контролем администратора Tuxedo SSO, так что Tuxedo SSO доверяет сущности аутентификации. Не рекомендуется использовать сущность аутентификации, которую администратор Tuxedo SSO не может контролировать.

Поставщик канала аутентификации предоставляется как поставщик SPI, чтобы пользователи Tuxedo SSO могли реализовать своего собственного поставщика для соответствия своей среде. Tuxedo SSO предоставляет своего поставщика по умолчанию, называемого поставщиком канала аутентификации HTTP, который использует HTTP для связи с объектом аутентификации.

Если пользователь Tuxedo SSO хочет использовать поставщика канала аутентификации HTTP, ему необходимо знать контракт между Tuxedo SSO и объектом аутентификации, состоящий из следующих двух частей.

Запрос/ответ на делегирование аутентификации

Tuxedo SSO отправляет запрос аутентификации субъекту аутентификации.

Уведомление о результате аутентификации/ACK

Объект аутентификации уведомляет Tuxedo SSO о результате аутентификации.

Запрос/ответ на делегирование аутентификации состоит из следующих сообщений.

Запрос на делегирование аутентификации

Запрос отправляется из Tuxedo SSO в объект аутентификации с просьбой выполнить аутентификацию пользователя с помощью AD.

ПОСТ [прием_делегации]

- Заголовки

Имя	Ценить	Описание
Тип контента	приложение/json	Тело сообщения имеет формат json.
Авторизация	Предъявитель [токен]	[Токен] используется, когда объект аутентификации уведомляет Tuxedo SSO о результате аутентификации.

- Параметры

Тип	Имя	Описание
Путь	прием_делегации	Конечная точка, предоставленная субъектом аутентификации для получения запроса на делегирование

- Тело

Имя	Описание
login_hint	Он сообщает субъекту аутентификации, кто аутентифицирован AD.

Имя	Описание
	По умолчанию это "имя пользователя" пользователя. Это поле является обязательным и определено стандартным документом CIBA.
объем	Он сообщает, в каких областях субъект аутентификации получает согласие от аутентифицированного пользователя. Это поле является обязательным и определено стандартным документом CIBA.
требуется_согласие	Показывает, необходимо ли субъекту аутентификации получить согласие от аутентифицированного пользователя на область действия. Это поле обязательно.
связывающее_сообщение	Его значение должно отображаться в пользовательском интерфейсе как CD, так и AD, чтобы пользователь мог понять, что аутентификация AD инициируется CD. Это поле является необязательным и было определено стандартным документом CIBA.
acr_values	Он сообщает запрашивающему Authentication Context Class Reference из CD. Это поле является необязательным и было определено стандартным документом CIBA.

Ответ на делегирование аутентификации

Ответ возвращается от объекта аутентификации в Tuxedo SSO для уведомления о том, что объект аутентификации получил запрос аутентификации от Tuxedo SSO.

- Ответы

HTTP-код статуса	Описание
201	Он уведомляет Tuxedo SSO о получении запроса на делегирование аутентификации.

Уведомление о результате аутентификации/ACK состоит из следующих сообщений.

Уведомление о результате аутентификации

Объект аутентификации отправляет результат запроса аутентификации в Tuxedo SSO.

POST /realms/[realm]/protocol/openid-connect/ext/ciba/auth/callback

- Заголовки

Имя	Ценить	Описание
Тип контента	приложение/json	Тело сообщения имеет формат json.
Авторизация	Предъявитель [токен]	[Токен] должен быть тем, который субъект аутентификации получил от Tuxedo SSO в запросе делегирования аутентификации.

- Параметры

Тип	Имя	Описание
Путь	область	Имя области

- Тело

Имя	Описание
статус	Он сообщает результат аутентификации пользователя с помощью AD. Он должен иметь один из следующих статусов. SUCCEED: Аутентификация с помощью AD успешно завершена. UNAUTHORIZED: Аутентификация с помощью AD не завершена. CANCELLED: Аутентификация с помощью AD отменена пользователем.

Результат аутентификации ACK

Ответ возвращается от Tuxedo SSO к объекту аутентификации, чтобы уведомить Tuxedo SSO о получении результата аутентификации пользователя AD от объекта аутентификации.

- Ответы

HTTP-код статуса	Описание
200	Он уведомляет субъект аутентификации о получении уведомления о результате аутентификации.

Поставщик распознавателя пользователей
Даже если пользователь один и тот же, его представление может отличаться в каждом CD, Tuxedo SSO и объекте аутентификации.

Чтобы CD, Tuxedo SSO и объект аутентификации распознавали одного и того же пользователя, этот поставщик User Resolver преобразует их собственные представления пользователей между собой.

User Resolver Provider предоставляется как поставщик SPI, чтобы пользователи Tuxedo SSO могли реализовать своего собственного поставщика для соответствия своей среде. Tuxedo SSO предоставляет своего поставщика по умолчанию, который называется Default User Resolver Provider, который имеет следующие характеристики.

- Поддерживается только login_hintпараметр, используемый по умолчанию.
- usernameUserModel в Tuxedo SSO используется для представления пользователя на CD, Tuxedo SSO и сущности аутентификации.

OIDC Выйти

OIDC имеет четыре спецификации, относящиеся к механизмам выхода из системы:

1. Управление сеансом
2. Выход, инициированный RP
3. Выход из переднего канала
4. Выход из обратного канала

Опять же, поскольку все это описано в спецификации OIDC, мы дадим здесь лишь краткий обзор.

Управление сеансом
Это выход из браузера. Приложение регулярно получает информацию о состоянии сеанса от Tuxedo SSO. Когда сеанс завершается в Tuxedo SSO, приложение это замечает и запускает собственный выход.

Выход, инициированный RP

Это также выход из системы на основе браузера, где выход начинается с перенаправления пользователя на определенную конечную точку в Tuxedo SSO. Это перенаправление обычно происходит, когда пользователь нажимает на ссылку Log Outна странице какого-либо приложения, которое ранее использовало Tuxedo SSO для аутентификации пользователя.

После того, как пользователь перенаправлен на конечную точку выхода из системы, Tuxedo SSO отправит клиентам запросы на выход из системы, чтобы позволить им аннулировать свои локальные сеансы пользователя, и потенциально перенаправит пользователя на некоторый URL-адрес после завершения процесса выхода из системы. Пользователю может быть дополнительно предложено подтвердить выход из системы, если id_token_hintпараметр не использовался. После выхода из системы пользователь автоматически перенаправляется на указанный post_logout_redirect_uri, если он указан в качестве параметра. Обратите внимание, что вам необходимо включить либо параметр client_id, либо id_token_hint, если post_logout_redirect_uriвключен . Также post_logout_redirect_uriпараметр должен соответствовать одному из Valid Post Logout Redirect URIsуказанных в конфигурации клиента.

В зависимости от конфигурации клиента запросы на выход могут быть отправлены клиентам через фронт-канал или через обратный канал. Для клиентов браузера frontend, которые полагаются на управление сеансами, описанное в предыдущем разделе, Tuxedo SSO не нужно отправлять им запросы на выход; эти клиенты автоматически определяют, что сеанс SSO в браузере завершен.

Выход из переднего канала

Чтобы настроить клиентов на получение запросов на выход из системы через front-channel, посмотрите на настройку клиента Front-Channel Logout . При использовании этого метода учтите следующее:

- Запросы на выход, отправляемые Tuxedo SSO клиентам, опираются на браузер и встроенные данные iframes, которые отображаются на странице выхода.

- Поскольку выход из системы на переднем канале основан на iframes, на него могут влиять политики безопасности контента (CSP), и запросы на выход могут блокироваться.
- Если пользователь закрывает браузер до отображения страницы выхода из системы или до того, как запросы на выход из системы фактически отправляются клиентам, его сеансы на клиенте могут не быть аннулированы.

Рассмотрите возможность использования функции выхода из системы по обратному каналу, поскольку она обеспечивает более надежный и безопасный подход к выходу пользователей из системы и завершению их сеансов на клиентах.

Если клиент не включен с выходом через передний канал, то Tuxedo SSO сначала попытается отправить запросы на выход через обратный канал, используя Back-Channel Logout URL . Если не определено, сервер вернется к использованию Admin URL .

Выход из обратного канала
Это не браузерный выход из системы, который использует прямую обратную связь между Tuxedo SSO и клиентами. Tuxedo SSO отправляет HTTP-запрос POST, содержащий токен выхода, всем клиентам, вошедшим в Tuxedo SSO. Эти запросы отправляются на зарегистрированные URL-адреса выхода из системы обратного канала в Tuxedo SSO и должны инициировать выход из системы на стороне клиента.

Конечные точки OIDC URI сервера Tuxedo SSO

Ниже приведен список конечных точек OIDC, которые публикует Tuxedo SSO. Эти конечные точки могут использоваться, когда клиентский адаптер, не являющийся Tuxedo SSO, использует OIDC для связи с сервером аутентификации. Все они являются относительными URL-адресами. Корень URL-адреса состоит из протокола HTTP(S), имени хоста и, опционально, пути: Например

https://localhost:8080

/realms/{имя-реальности}/протокол/openid-connect/auth

Используется для получения временного кода в потоке кода авторизации или получения токенов с использованием неявного потока, прямых грантов или клиентских грантов.

/realms/{имя-реальности}/протокол/openid-connect/токен

Используется потоком кода авторизации для преобразования временного кода в токен.

/realms/{имя-реальности}/протокол/openid-connect/logout

Используется для выполнения выхода из системы.

/realms/{имя-реальности}/протокол/openid-connect/userinfo

Используется для сервиса информации о пользователе, описанного в спецификации OIDC.

/realms/{имя-области}/протокол/openid-connect/revoke

Используется для отзыва токена OAuth 2.0, описанного в RFC7009 .

/realms/{имя-области}/протокол/openid-connect/certs

Используется для набора веб-ключей JSON (JWKS), содержащего открытые ключи, используемые для проверки любого веб-токена JSON (jwks_uri)

/realms/{имя-реальности}/протокол/openid-connect/auth/устройство

Используется для предоставления авторизации устройства с целью получения кода устройства и кода пользователя.

/realms/{имя-реальности}/протокол/openid-connect/ext/ciba/auth

Это конечная точка URL для инициированного клиентом гранта аутентификации обратного канала для получения auth_req_id, который идентифицирует запрос аутентификации, сделанный клиентом.

/realms/{имя-реальности}/протокол/openid-connect/logout/backchannel-logout

Это конечная точка URL для выполнения выходов из системы по обратному каналу, описанных в спецификации OIDC.

Во всех этих случаях замените {realm-name} на имя области.

САМЛ

SAML 2.0 — это спецификация, похожая на OIDC, но более зрелая. Она произошла от спецификаций SOAP и обмена сообщениями веб-сервисов, поэтому, как правило, более многословна, чем OIDC. SAML 2.0 — это протокол аутентификации, который обменивается XML-документами между серверами аутентификации и приложениями. Для проверки запросов и ответов используются XML-подписи и шифрование.

В целом SAML реализует два варианта использования.

Первый вариант использования — приложение, которое запрашивает у сервера Tuxedo SSO аутентификацию пользователя. После успешного входа приложение получит XML-документ. Этот документ содержит утверждение SAML, которое определяет атрибуты пользователя. Область цифровой подписью подписывает документ, содержащий информацию о доступе (например, сопоставления ролей пользователей), которую приложения используют для определения ресурсов, к которым пользователям разрешен доступ в приложении.

Второй вариант использования — клиент, получающий доступ к удаленным сервисам. Клиент запрашивает утверждение SAML от Tuxedo SSO для вызова удаленных сервисов от имени пользователя.

SAML-привязки

Tuxedo SSO поддерживает три типа привязки.

Перенаправление привязки
Привязка перенаправления использует ряд URI перенаправления браузера для обмена информацией.

1. Пользователь подключается к приложению с помощью браузера. Приложение обнаруживает, что пользователь не аутентифицирован.
2. Приложение генерирует XML-документ запроса аутентификации и кодирует его как параметр запроса в URI. URI используется для перенаправления на сервер Tuxedo SSO. В зависимости от ваших настроек приложение также может подписать XML-документ цифровой подписью и включить подпись как параметр запроса в URI перенаправления на Tuxedo SSO. Эта подпись используется для проверки клиента, который отправляет запрос.
3. Браузер перенаправляет на Tuxedo SSO.
4. Сервер извлекает XML-документ запроса на аутентификацию и при необходимости проверяет цифровую подпись.
5. Пользователь вводит свои учетные данные для аутентификации.
6. После аутентификации сервер генерирует ответный документ аутентификации XML. Документ содержит утверждение SAML, которое содержит метаданные о пользователе, включая имя, адрес, адрес электронной почты и любые сопоставления ролей, которые есть у пользователя. Документ обычно имеет цифровую подпись с использованием подписей XML, а также может быть зашифрован.
7. Документ ответа аутентификации XML кодируется как параметр запроса в URI перенаправления. URI возвращает браузер обратно в приложение. Цифровая подпись также включена как параметр запроса.
8. Приложение получает URI перенаправления и извлекает XML-документ.
9. Приложение проверяет подпись области, чтобы убедиться, что оно получает действительный ответ аутентификации. Информация внутри утверждения SAML используется для принятия решений о доступе или отображения данных пользователя.

POST-связывание

Привязка POST похожа на привязку Redirect , но привязка POST обменивается XML-документами с помощью запросов POST вместо запросов GET.
Привязка POST использует JavaScript, чтобы браузер отправлял запрос POST на

сервер или приложение Tuxedo SSO при обмене документами. HTTP отвечает HTML-документом, содержащим HTML-форму со встроенным JavaScript. Когда страница загружается, JavaScript автоматически вызывает форму.

Привязка POST рекомендуется из-за двух ограничений:

- Безопасность — При перенаправлении привязки ответ SAML является частью URL. Это менее безопасно, так как можно зафиксировать ответ в журналах.
- Размер — Отправка документа в полезной нагрузке HTTP обеспечивает больше возможностей для больших объемов данных, чем при использовании ограниченного URL.

ЕСП
Enhanced Client or Proxy (ECP) — это профиль SAML v.2.0, который позволяет обмениваться атрибутами SAML вне контекста веб-браузера. Он часто используется клиентами на основе REST или SOAP.

Конечные точки URI SAML сервера Tuxedo SSO

Tuxedo SSO имеет одну конечную точку для всех запросов SAML.

http(s)://authserver.host/realms/{realm-name}/protocol/saml

Все привязки используют эту конечную точку.

OpenID Connect в сравнении с SAML

Ниже перечислен ряд факторов, которые следует учитывать при выборе протокола.

Для большинства целей Tuxedo SSO рекомендует использовать OIDC.

ОИДК

- OIDC специально разработан для работы с Интернетом.
- OIDC подходит для приложений HTML5/JavaScript, поскольку его проще реализовать на стороне клиента, чем SAML.

- Токены OIDC имеют формат JSON, что упрощает их использование в Javascript.
- OIDC имеет функции, упрощающие реализацию безопасности. Например, см. трюк с iframe , который спецификация использует для определения статуса входа пользователя.

САМЛ

- SAML разработан как слой, работающий поверх Интернета.
- SAML может быть более подробным, чем OIDC.
- Пользователи выбирают SAML вместо OIDC, поскольку считают его более зрелым.
- Пользователи выбирают SAML вместо существующих приложений OIDC, защищенных с его помощью.

Аутентификация Docker Registry v2

Docker-аутентификация отключена по умолчанию. Чтобы включить docker-аутентификацию, см. руководство Включение и отключение функций .

Docker Registry V2 Authentication — это протокол, похожий на OIDC, который аутентифицирует пользователей по реестрам Docker. Реализация этого протокола Tuxedo SSO позволяет клиентам Docker использовать сервер аутентификации Tuxedo SSO для аутентификации по реестру. Этот протокол использует стандартные механизмы токенов и подписей, но он отличается от настоящей реализации OIDC. Он отличается использованием очень специфичного формата JSON для запросов и ответов, а также сопоставлением имен репозиториев и разрешений с механизмом области действия OAuth.

Процесс аутентификации Docker

Поток аутентификации описан в документации API Docker . Ниже приведено краткое изложение с точки зрения сервера аутентификации Tuxedo SSO:

- Выполните docker login.

- Клиент Docker запрашивает ресурс из реестра Docker. Если ресурс защищен и в запросе нет токена аутентификации, сервер реестра Docker отвечает HTTP-сообщением 401 с некоторой информацией о требуемых разрешениях и местоположении сервера авторизации.
- Клиент Docker создает запрос аутентификации на основе HTTP-сообщения 401 из реестра Docker. Клиент использует локально кэшированные учетные данные (из команды docker login) как часть запроса HTTP Basic Authentication к серверу аутентификации Tuxedo SSO.
- Сервер аутентификации Tuxedo SSO пытается аутентифицировать пользователя и возвращает тело JSON, содержащее токен Bearer в стиле OAuth.
- Клиент Docker получает токен носителя из ответа JSON и использует его в заголовке авторизации для запроса защищенного ресурса.
- Реестр Docker получает новый запрос на защищенный ресурс с токеном от сервера Tuxedo SSO. Реестр проверяет токен и предоставляет доступ к запрошенному ресурсу (если применимо).

Tuxedo SSO не создает сеанс SSO браузера после успешной аутентификации с протоколом Docker. Сеанс SSO браузера не использует протокол Docker, поскольку он не может обновлять токены или получать статус токена или сеанса с сервера Tuxedo SSO; поэтому сеанс SSO браузера не нужен. Более подробную информацию см. в разделе о временных сеансах .

Конечные точки URI сервера аутентификации Tuxedo SSO Docker Registry v2

Tuxedo SSO имеет одну конечную точку для всех запросов Docker auth v2.

http(s)://authserver.host/realms/{realm-name}/protocol/docker-v2/auth

Управление доступом к консоли администратора

Каждая область, созданная на Tuxedo SSO, имеет выделенную консоль администратора, из которой можно управлять этой областью. masterОбласть — это особая область, которая позволяет администраторам управлять более чем одной областью в системе. Вы также можете определить детальный доступ для

пользователей в разных областях для управления сервером. В этой главе рассматриваются все сценарии для этого.

Главный контроль доступа к области

Область masterв Tuxedo SSO является особой областью и обрабатывается иначе, чем другие области. Пользователям в masterобласти Tuxedo SSO может быть предоставлено разрешение на управление нулем или более областей, развернутых на сервере Tuxedo SSO. При создании области Tuxedo SSO автоматически создает различные роли, которые предоставляют детальные разрешения на доступ к этой новой области. Доступ к консоли администратора и конечным точкам REST администратора можно контролировать, сопоставляя эти роли с пользователями в masterобласти. Можно создать несколько суперпользователей, а также пользователей, которые могут управлять только определенными областями.

Глобальные роли

В сфере есть две роли уровня области master. Это:

- админ
- создать-царство

Пользователи с этой adminролью являются суперпользователями и имеют полный доступ к управлению любой областью на сервере. Пользователи ccreate-realm ролью могут создавать новые области. Им будет предоставлен полный доступ к любой новой области, которую они создают.

Роли, специфичные для сферы

Администратору в masterобласти могут быть предоставлены привилегии управления одной или несколькими другими областями в системе. Каждая область в Tuxedo SSO представлена клиентом в masterобласти. Имя клиента — <realm name>-realm. У каждого из этих клиентов определены роли на уровне клиента, которые определяют различные уровни доступа для управления отдельной областью.

Доступны следующие роли:

- область просмотра
- просмотр-пользователей
- просмотр-клиентов
- просмотр событий
- управлять-областью
- управлять пользователями
- создать-клиент
- управлять клиентами
- управлять событиями
- просмотр-идентификаторов-провайдеров
- управлять-идентификационными-провайдерами
- олицетворение

Назначьте нужные роли своим пользователям, и они смогут использовать только эту конкретную часть консоли администрирования.

> Администраторы с этой `manage-users`ролью смогут назначать только те роли администратора, которые есть у них самих. Таким образом, если у администратора есть роль `manage-users`, но нет `manage-realm`роли, он не сможет назначить эту роль.

Выделенные консоли администратора области

Каждая область имеет выделенную консоль администратора, доступ к которой можно получить, перейдя по адресу url /admin/{realm-name}/console. Пользователям в этой области можно предоставить разрешения на управление областью, назначив определенные сопоставления ролей пользователей.

Каждая область имеет встроенный клиент, называемый realm-management. Вы можете просмотреть этого клиента, перейдя в Clientsлевый пункт меню вашей области. Этот клиент определяет роли на уровне клиента, которые указывают разрешения, которые могут быть предоставлены для управления областью.

- область просмотра
- просмотр-пользователей
- просмотр-клиентов
- просмотр событий
- управлять-областью
- управлять пользователями
- создать-клиент
- управлять клиентами
- управлять событиями
- просмотр-идентификаторов-провайдеров
- управлять-идентификационными-провайдерами
- олицетворение

Назначьте нужные роли своим пользователям, и они смогут использовать только эту конкретную часть консоли администрирования.

Тонкие разрешения администратора

> Fine Grain Admin Permissions — это **предварительная версия** , которая не поддерживается полностью. Эта функция отключена по умолчанию.
>
> Чтобы включить запустите сервер с помощью `--features=preview`или`--features=admin-fine-grained-authz`

Иногда роли, такие как manage-realmили manage-usersслишком грубые, и вы хотите создать ограниченные учетные записи администратора с более мелкими разрешениями. Tuxedo SSO позволяет вам определять и назначать политики ограниченного доступа для управления областью. Такие вещи, как:

- Управление одним конкретным клиентом
- Управление пользователями, принадлежащими к определенной группе

- Управление членством в группе
- Ограниченное управление пользователями.
- Тонкий контроль над подражанием
- Возможность назначать пользователям определенный ограниченный набор ролей.
- Возможность назначить составной роли определенный ограниченный набор ролей.
- Возможность назначения определенного ограниченного набора ролей в рамках деятельности клиента.
- Новые общие политики для просмотра и управления пользователями, группами, ролями и клиентами.

Следует отметить несколько важных моментов относительно детальных прав администратора:

- Разрешения администратора с мелкозернистой структурой были реализованы поверх служб авторизации . Настоятельно рекомендуется ознакомиться с этими функциями, прежде чем углубляться в разрешения с мелкозернистой структурой.
- Разрешения Fine Granite доступны только в выделенных консолях администратора и администраторах, определенных в этих областях. Вы не можете определить разрешения Cross Realm Fine Granite.
- Для предоставления дополнительных разрешений используются разрешения Fine Granite. Вы не можете переопределить поведение по умолчанию встроенных ролей администратора.

Управление одним конкретным клиентом

Давайте сначала рассмотрим разрешение администратору управлять одним клиентом и только одним клиентом. В нашем примере у нас есть область с именем testи клиент с именем sales-application. В области testмы дадим пользователю в этой области разрешение только на управление этим приложением.

> Вы не можете делать межобластные разрешения с высокой степенью детализации. Администраторы в `master`области ограничены предопределенными ролями администраторов, определенными в предыдущих главах.

Настройка разрешения

Первое, что мы должны сделать, это войти в консоль администратора, чтобы мы могли настроить разрешения для этого клиента. Мы переходим в раздел управления клиента, для которого мы хотим определить детальные разрешения.

Управление клиентами

Вы должны увидеть пункт меню вкладки под названием Permissions. Нажмите на эту вкладку.

Вкладка «Клиентские разрешения»

По умолчанию каждый клиент не может выполнять точную настройку разрешений. Поэтому включите Permissions Enabledпереключатель, чтобы инициализировать разрешения.

> Если вы установите `Permissions Enabled`переключатель в положение «Выкл.», будут удалены все разрешения, которые вы определили для этого клиента.

Вкладка «Клиентские разрешения»

При переключении Permissions Enabledна on он инициализирует различные объекты разрешений за кулисами с помощью Authorization Services . В этом примере нас интересует разрешение manageдля клиента. Щелчок по нему перенаправит вас к разрешению, которое обрабатывает manageразрешение для клиента. Все объекты авторизации содержатся на вкладке realm-managementклиента Authorization.

Разрешение на управление клиентом

При первой инициализации manageразрешение не имеет связанных с ним политик. Вам нужно будет создать одну, перейдя на вкладку политики. Чтобы быстро туда попасть, щелкните ссылку, Client detailsпоказанную на изображении выше. Затем щелкните вкладку политики.

На этой странице найдите Create client policукнопку, которую можно использовать для определения множества политик. Вы можете определить политику, связанную

с ролью или группой, или даже определить правила в JavaScript. Для этого простого примера мы создадим User Policy.

Политика пользователя

Эта политика будет соответствовать жестко закодированному пользователю в базе данных пользователей. В данном случае это пользователь sales-admin. Затем мы должны вернуться на страницу разрешений sales-applicationклиента manageи назначить политику объекту разрешения.

Назначить политику пользователя

Теперь у пользователя sales-adminесть разрешение на управление sales-applicationклиентом.

Нам осталось сделать еще одну вещь. Перейдите в Users, выберите sales-adminпользователя, затем перейдите на Role Mappingsвкладку и назначьте query-clientsроль пользователю.

Назначить запросы-клиенты

Зачем вам это нужно? Эта роль сообщает Admin Console, какие пункты меню отображать при sales-adminпосещении Admin Console. query-clientsРоль сообщает Admin Console, что она должна отображать клиентские меню для sales-adminпользователя.

ВАЖНО Если вы не установите query-clientsроль, ограниченные администраторы, например, sales-adminне будут видеть никаких пунктов меню при входе в консоль администратора.

Тестирование
Далее мы выходим из главной области и повторно входим в специальную консоль администратора для testобласти, используя в sales-adminкачестве имени пользователя. Она находится в разделе /admin/test/console.

Вход для администратора продаж

Теперь этот администратор может управлять этим одним клиентом.

Ограничить сопоставление ролей пользователей

Еще одна вещь, которую вы, возможно, захотите сделать, это ограничить набор ролей, которые администратор может назначать пользователю. Продолжая наш последний пример, давайте расширим набор разрешений пользователя 'sales-admin', чтобы он также мог контролировать, каким пользователям разрешен доступ к этому приложению. С помощью разрешений с точной детализацией мы можем включить его так, чтобы он sales-adminмог назначать только роли, которые предоставляют определенный доступ к sales-application. Мы также можем ограничить его так, чтобы администратор мог только сопоставлять роли и не выполнять никаких других типов администрирования пользователей.

Определены sales-applicationтри различные роли клиентов.

Роли в приложениях по продажам

Мы хотим, чтобы sales-adminпользователь мог сопоставлять эти роли с любым пользователем в системе. Первый шаг для этого — разрешить администратору сопоставлять роль. Если мы нажмем на роль , вы увидите, что для этой роли viewLeadsесть вкладка.Permissions

Просмотр вкладки разрешений ролей лидов

Если мы нажмем на эту вкладку и включим ее Permissions Enabled, вы увидите, что существует ряд действий, к которым мы можем применять политики.

Просмотр разрешений на лиды

Нас интересует map-role. Щелкните по этому разрешению и добавьте ту же политику пользователя, которая была создана в предыдущем примере.

Разрешение на использование ролей карты

Что мы сделали, так это сказали, что sales-adminмогут сопоставлять viewLeadsроль. Чего мы не сделали, так это указали, каким пользователям администратор также может сопоставлять эту роль. Чтобы сделать это, мы должны перейти в Usersраздел консоли администратора для этой области. Нажатие на Usersлевый пункт меню переносит нас в пользовательский интерфейс области. Вы должны увидеть вкладку Permissions. Нажмите на нее и включите ее.

Разрешения пользователей

Разрешение, которое нас интересует, — это map-roles. Это ограничительная политика, которая позволяет только администраторам сопоставлять роли с пользователем. Если мы нажмем на map-rolesразрешение и снова добавим политику пользователя, которую мы создали для этого, мы sales-admincможем сопоставлять роли с любым пользователем.

Последнее, что нам осталось сделать, это добавить view-usersроль в sales-admin. Это позволит администратору просматривать пользователей в области, sales-applicationв которую он хочет добавить роли.

Добавить просмотр-пользователей

Тестирование
Далее мы выходим из главной области и повторно входим в специальную консоль администратора для testобласти, используя в sales-adminкачестве имени пользователя. Она находится в разделе /admin/test/console.

Вы увидите, что теперь sales-adminможно просматривать пользователей в системе. Если вы выберете одного из пользователей, вы увидите, что каждая страница сведений о пользователе доступна только для чтения, за исключением Role Mappingsвкладки . Перейдя на эту вкладку, вы обнаружите, что нет Availableролей, которые администратор мог бы сопоставить пользователю, за исключением случаев, когда мы просматриваем sales-applicationроли.

Назначить viewLeads

Мы только указали, что sales-adminможем сопоставить viewLeadsроль.

Ярлык для ролей карты клиента
Было бы утомительно, если бы нам пришлось делать это для каждой опубликованной клиентской роли sales-application. Чтобы упростить задачу, есть способ указать, что администратор может сопоставить любую роль, определенную клиентом. Если мы снова войдем в консоль администратора в качестве администратора нашей главной области и вернемся на sales-applicationстраницу разрешений, вы увидите map-rolesразрешение.

Разрешение на использование клиентской карты ролей

Если вы предоставите администратору доступ к этому конкретному разрешению, этот администратор сможет сопоставить любую роль, определенную клиентом.

Полный список разрешений

Вы можете сделать гораздо больше с разрешениями точного гранулярного уровня, помимо управления определенным клиентом или определенными ролями клиента. В этой главе определяется весь список типов разрешений, которые могут быть описаны для области.

Роль

При переходе на Permissionsвкладку определенной роли вы увидите список этих типов разрешений.

карта-роль

> Политики, которые решают, может ли администратор сопоставить эту роль пользователю. Эти политики только указывают, что роль может быть сопоставлена пользователю, а не то, что администратору разрешено выполнять задачи сопоставления ролей пользователей. Администратор также должен иметь разрешения на управление или сопоставление ролей.
> См. Разрешения пользователей для получения дополнительной информации.

карта-роль-композит

> Политики, которые решают, может ли администратор сопоставить эту роль как составную с другой ролью. Администратор может определить роли для клиента, если он должен управлять разрешениями для этого клиента, но он не сможет добавлять составные роли к этим ролям, если у него нет привилегий map-role-compositeдля роли, которую он хочет добавить как составную.

карта-роль-клиент-область

> Политики, которые решают, может ли администратор применять эту роль к области действия клиента. Даже если администратор может управлять

клиентом, у него не будет разрешения создавать токены для этого клиента, содержащие эту роль, если эта привилегия не предоставлена.

Клиент

При переходе на Permissionsвкладку конкретного клиента вы увидите список этих типов разрешений.

вид

Политики, которые определяют, может ли администратор просматривать конфигурацию клиента.

управлять

Политики, которые решают, может ли администратор просматривать и управлять конфигурацией клиента. С этим связаны некоторые проблемы, связанные с тем, что привилегии могут быть непреднамеренно утечек. Например, администратор может определить сопоставителя протоколов, который жестко закодирует роль, даже если у администратора нет привилегий для сопоставления роли с областью действия клиента. В настоящее время это ограничение сопоставителей протоколов, поскольку у них нет способа назначать им индивидуальные разрешения, как это делают роли.

настроить

Сокращенный набор привилегий для управления клиентом. Это похоже на manageобласть действия, за исключением того, что администратору не разрешено определять сопоставители протоколов, изменять шаблон клиента или область действия клиента.

карта-роли

Политики, которые решают, может ли администратор сопоставить любую роль, определенную клиентом, с пользователем. Это кратчайший путь, простая в использовании функция, позволяющая избежать необходимости определять политики для каждой роли, определенной клиентом.

карта-роли-композит

Политики, которые решают, может ли администратор сопоставить любую роль, определенную клиентом как составную, с другой ролью. Это кратчайший путь, простая в использовании функция, позволяющая избежать необходимости определять политики для каждой роли, определенной клиентом.

карта-роли-клиент-область

Политики, которые решают, может ли администратор сопоставить любую роль, определенную клиентом, с областью действия другого клиента. Это кратчайший путь, простая в использовании функция, позволяющая избежать необходимости определять политики для каждой роли, определенной клиентом.

Пользователи

При переходе на Permissionsвкладку «Все пользователи» вы увидите список этих типов разрешений.

вид

Политики, которые определяют, может ли администратор просматривать всех пользователей в области.

управлять

Политики, которые определяют, может ли администратор управлять всеми пользователями в области. Это разрешение предоставляет администратору привилегию выполнять сопоставление ролей пользователей, но не указывает, какие роли администратору разрешено сопоставлять. Вам нужно будет определить привилегию для каждой роли, которую вы хотите, чтобы администратор мог сопоставлять.

карта-роли

Это подмножество привилегий, предоставляемых manageобластью действия. В этом случае администратору разрешено только сопоставлять роли. Администратору не разрешено выполнять какие-либо другие операции по управлению пользователями. Кроме того, как и manage, роли, которые

администратору разрешено применять, должны быть указаны для каждой роли или для каждого набора ролей, если речь идет о клиентских ролях.

управлять-групповым-членством

Аналогично, за map-rolesисключением того, что это относится к членству в группах: к каким группам пользователь может быть добавлен или удален из них. Эти политики просто предоставляют администратору разрешение на управление членством в группах, а не к тем группам, в которых администратору разрешено управлять членством. Вам придется указать политики для manage-membersразрешения каждой группы.

выдавать себя за другого

Политики, которые решают, разрешено ли администратору выдавать себя за других пользователей. Эти политики применяются к атрибутам администратора и сопоставлениям ролей.

выдаваемый за пользователя

Политики, которые решают, какие пользователи могут быть выданы за другого пользователя. Эти политики будут применяться к пользователю, который выдается за другого пользователя. Например, вы можете определить политику, которая запретит кому-либо выдавать себя за пользователя с правами администратора.

Группа

При переходе на Permissionsвкладку определенной группы вы увидите список этих типов разрешений.

вид

Политики, которые определяют, может ли администратор просматривать информацию о группе.

управлять

Политики, которые определяют, может ли администратор управлять конфигурацией группы.

просмотр-участников

: Политики, которые определяют, может ли администратор просматривать данные пользователей членов группы.

управлять-членами

: Политики, которые определяют, может ли администратор управлять пользователями, принадлежащими к этой группе.

управлять-членством

: Политики, которые решают, может ли администратор изменить членство в группе. Добавлять или удалять участников из группы.

Управление организациями

При интеграции с третьей стороной, например клиентом или деловым партнером, вам может потребоваться управлять их идентификационными данными отдельно от других и создать единый и безопасный интерфейс во всей вашей бизнес-экосистеме при их взаимодействии с областью.

В области организация представляет эти третьи стороны, так что администратор области или организации может управлять всем жизненным циклом ее членов, а также тем, как они аутентифицируются и авторизуются в области, для каждой организации.

Организация является точкой входа для начала использования возможностей IAM Tuxedo SSO для решения также бизнес-задач (B2B). Она обеспечивает многопользовательскую аренду в рамках области, чтобы пользователи могли иметь доступ к защищенным ресурсам из области, но с более ограниченным и контролируемым контекстом. Организация, к которой они принадлежат.

Tuxedo SSO Organizations — это функция, которая обеспечивает поддержку организаций в Tuxedo SSO. На данный момент она предоставляет некоторые основные возможности, необходимые для управления организациями, такие как:

- Управление участниками
- Привлечение членов организации с использованием ссылок-приглашений
- Привлечение членов организации путем объединения их личностей посредством посредничества в области идентификации
- Вход в систему с приоритетом идентификации и действия, специфичные для организации, при аутентификации в рамках организации
- Распространение претензий, специфичных для организации, на приложения с помощью токенов для целей авторизации

Поддержка организаций в Tuxedo SSO

Чтобы использовать организации, вам необходимо включить эту функцию для текущей области.

Процедура

1. Нажмите «Настройки области» в меню.
2. Включите « Организации » .
3. Нажмите «Сохранить».

Организации, оказывающие поддержку

После включения функции вы сможете управлять организациями через раздел «Организации» , доступный в меню.

Управление организацией

В разделе «Организации» вы можете управлять всеми организациями в вашей сфере.

Управление организациями

Создание организации

Процедура

1. Нажмите Создать организацию .

Создание организации

Организация имеет следующие настройки:

Имя

Удобное для пользователя название организации. Название уникально в пределах области.

Псевдоним

Псевдоним для этой организации, используемый для внутренней ссылки на организацию. Псевдоним уникален в пределах области и должен быть дружественным к URL, поэтому символы, обычно не разрешенные в URL, не будут разрешены в псевдониме. Если не задано, Tuxedo SSO попытается использовать имя в качестве псевдонима. Если имя не дружественно к URL, вы получите сообщение об ошибке и вам будет предложено указать псевдоним. После определения псевдоним нельзя будет изменить.

URL-адрес перенаправления

После завершения регистрации или принятия приглашения в организацию, отправленного по электронной почте, пользователь автоматически перенаправляется на указанный URL-адрес перенаправления. Если оставить пустым, пользователь будет перенаправлен в консоль учетной записи по умолчанию.

Домены

Набор из одного или нескольких доменов, принадлежащих этой организации. Домен не может совместно использоваться разными организациями в пределах области.

Описание

Поле для свободного текста для описания организации.

После создания организации вы сможете управлять дополнительными настройками, которые описаны в следующих разделах:

- Управление атрибутами
- Управление участниками
- Управление поставщиками удостоверений

Понимание организационных доменов

При управлении организацией домен, связанный с организацией, играет важную роль в том, как члены организации проходят аутентификацию в сфере и как проверяются их профили.

Одна из ключевых ролей домена — помочь идентифицировать организации, членом которых является пользователь. Просматривая адрес электронной почты, Tuxedo SSO сопоставит соответствующую организацию, использующую тот же домен, и в конечном итоге изменит поток аутентификации на основе требований организации.

Домен также позволяет организациям принудительно запретить пользователям использовать в своих электронных письмах домен, отличный от связанного с организацией. Это ограничение особенно полезно, когда пользователи и их удостоверения объединены с поставщиками удостоверений, связанными с организацией, и вы хотите принудительно указать определенный домен электронной почты для их адресов электронной почты.

Отключение организации

Чтобы отключить организацию, переключите «Включено» в положение «Выкл.» .

Отключение организации

Если организация отключена, вы по-прежнему можете управлять ею через интерфейсы управления, но члены организации не смогут проходить аутентификацию в этой области, в том числе через поставщиков удостоверений, связанных с организацией, поскольку они также автоматически отключены.

Однако неуправляемые члены организации по-прежнему могут проходить аутентификацию в сфере, поскольку они также являются пользователями сферы, но токены не будут содержать метаданные об их отношениях с организацией, которая отключена.

Более подробную информацию об управляемых и неуправляемых пользователях см. в разделе Управляемые и неуправляемые участники .

Удаление организации

Чтобы удалить организацию, нажмите кнопку « Удалить » для соответствующей организации на странице со списком или при редактировании организации.

Удаление организации

При удалении организации все данные, связанные с ней, будут удалены, включая всех управляемых участников.

Неуправляемые пользователи и поставщики удостоверений остаются в сфере, но они больше не связаны с организацией.

Более подробную информацию об управляемых и неуправляемых пользователях см. в разделе Управляемые и неуправляемые участники .

Управление атрибутами

Администратор может хранить дополнительные метаданные об организации с помощью атрибутов. Атрибут организации — это пара ключ/значение, которая может содержать несколько строковых значений.

Для этого перейдите на вкладку Атрибуты и задайте любой нужный атрибут, указав ключ и значение.

Чтобы указать несколько значений для одного и того же атрибута и ключа, просто добавьте еще один атрибут с тем же ключом, но с другим значением.

Управление атрибутами организации

Управляющие члены

Член организации — это, по сути, пользователь области, но со ссылкой на одну организацию. Они логически отделены от других пользователей в области, так что вы точно знаете, какие пользователи принадлежат к организации.

Существуют различные способы добавления или приема члена в организацию:

- Добавление существующего пользователя области в качестве участника
- Через поставщика удостоверений, связанного с организацией
- Отправка приглашения на создание новой учетной записи
- Отправка приглашения существующему пользователю присоединиться к организации

Став членом организации, вы можете управлять своей учетной записью так же, как и любой обычной учетной записью в сфере, перейдя в раздел «Пользователи» в меню.

Однако вы можете сузить круг пользователей до тех, кто связан с организацией, перейдя на вкладку «Участники» при управлении организацией. На этой вкладке у вас есть список всех членов организации и действия по добавлению новых членов, а также по редактированию и удалению существующих.

Управление членами организации

Управляемые и неуправляемые члены

При управлении участниками учитывайте, как их отношения с организацией влияют на жизненный цикл их учетных записей. Участники могут присоединиться к организации через разные потоки, и каждый поток указывает на прочность связи между их учетными записями и организацией.

Существует два типа участников:

- Удалось
- Неуправляемый

Управляемые участники — это те, которыми управляет организация, и они не могут существовать вне своей организации. Например, рассмотрим учетную запись, созданную через поставщика удостоверений, связанного с организацией. Эта учетная запись не принадлежит области, поскольку она была объединена с организацией. В этом случае единственным источником истины для удостоверения является организация, и ее жизненный цикл контролируется организацией. Если вы удалите организацию или участника, учетная запись также будет удалена из области.

С другой стороны, неуправляемые участники — это те, кто может существовать без организации. Например, при добавлении существующего пользователя области в организацию учетная запись принадлежит в первую очередь области и в конечном итоге связана с организацией. В этом случае удаление организации или участника не приведет к удалению учетной записи из области; область является единственным источником истины для идентификации.

Добавление существующего пользователя области в качестве участника

Существующий пользователь области может присоединиться к организации, выбрав этого пользователя из списка и добавив его в организацию.

Процедура

1. Нажмите Добавить участника .
2. Нажмите Добавить пользователя области .
3. Выберите одного или нескольких пользователей и нажмите «Добавить» , чтобы добавить их в организацию.

Добавление пользователя области

Как только пользователь становится членом организации, он может проходить аутентификацию в сфере так же, как и обычный пользователь, используя любые учетные данные, поддерживаемые сферой.

Приглашение пользователей

Администратор может отправлять пользователям электронные письма с просьбой присоединиться к организации.

Процедура

1. Нажмите Добавить участника .
2. Нажмите Пригласить участника .
3. Укажите адрес электронной почты
4. Нажмите «Отправить».

Приглашение участников

При желании вы также можете указать значения для полей «Имя» и «Фамилия» , чтобы сделать сообщение электронной почты более персонализированным, используя приветственное сообщение с именем и фамилией получателя.

Приглашение — это, по сути, электронное письмо со ссылкой, по которой человек должен щелкнуть, чтобы выполнить необходимые шаги для присоединения к организации. Эти шаги зависят от того, есть ли у человека уже учетная запись в сфере или необходимо создать новую учетную запись перед присоединением к организации.

Если адрес электронной почты соответствует существующему пользователю в определенной области, то действия, которые должен выполнить пользователь, в основном будут направлены на подтверждение намерения присоединиться к организации.

С другой стороны, если ни один пользователь не связан с указанным адресом электронной почты, шаги будут включать создание новой учетной записи через поток самостоятельной регистрации области. В этом случае пользователь вынужден предоставить тот же адрес электронной почты, который использовался для отправки приглашения.

Регистрация участников с использованием поставщика удостоверений

Организация может иметь собственного поставщика удостоверений в качестве единственного источника истины для своих удостоверений. В этом случае пользователи, объединенные с поставщиком удостоверений, автоматически добавляются в качестве членов организации.

Когда пользователи присоединяются к организации через поставщика удостоверений, связанного с организацией, они автоматически помечаются как управляемые участники. В этом случае они пройдут через потоки входа брокера, настроенные в области, и автоматически присоединятся к организации после успешной аутентификации.

Регистрация новых участников через поставщика удостоверений может осуществляться либо путем автоматического перенаправления пользователя к поставщику удостоверений организации, либо путем выбора поставщика удостоверений на странице входа.

В обоих случаях, как только пользователь предоставит адрес электронной почты, Tuxedo SSO попытается сопоставить организацию на основе домена электронной почты. В случае, если домен электронной почты соответствует организации, а поставщик удостоверений связан с тем же доменом и включена настройка Redirect when email domain matches , пользователь автоматически перенаправляется к поставщику удостоверений. После того, как пользователь проходит аутентификацию у поставщика удостоверений и завершает первый поток входа брокера, пользователь автоматически добавляется в качестве члена организации.

С другой стороны, если функция «Перенаправление при совпадении домена электронной почты» не включена, но поставщик удостоверений не настроен на « Скрыть на странице входа» , пользователь может выбрать поставщика удостоверений, а затем быть перенаправленным к поставщику удостоверений для продолжения процесса регистрации.

Более подробную информацию см. в разделе Управление поставщиками удостоверений .

Удаление участника

Вы можете удалить участника из организации.

В меню действий рядом с участником, которого вы хотите удалить, нажмите Удалить .

При удалении участника из организации помните, что пользователь может быть удален или не удален из области в зависимости от того, является ли этот пользователь управляемым или неуправляемым участником соответственно.

Более подробную информацию см. в разделе Управляемые и неуправляемые члены .

Управление поставщиками удостоверений

Организация может иметь собственного поставщика удостоверений в качестве единственного источника истины для своих удостоверений. В этом случае вы хотите настроить организацию для аутентификации пользователей с использованием поставщика удостоверений организации, объединить их удостоверения и, наконец, добавить их в качестве членов организации.

С организацией может быть связан один или несколько поставщиков удостоверений, что позволит им аутентифицировать своих пользователей из разных источников и применять различные ограничения к каждому из них.

Прежде чем вы сможете связать поставщика удостоверений с организацией, вы создаете организацию на уровне области из раздела Поставщики удостоверений в меню. Вы можете связать любой из встроенных поставщиков социальных сетей и удостоверений, доступных в области, с организацией.

Привязка поставщика удостоверений к организации

Поставщик удостоверений может быть связан с организацией на вкладке Поставщики удостоверений . Если поставщики удостоверений уже существуют, вы увидите их список и опции для поиска, редактирования или отмены связи с организацией.

Поставщики удостоверений организаций

Процедура

1. Нажмите ссылку поставщика удостоверений
2. Выберите поставщика удостоверений

3. Установите соответствующие настройки
4. Нажмите «Сохранить».

Связывание поставщика удостоверений

Поставщик удостоверений имеет следующие настройки:

Поставщик удостоверений

Поставщик удостоверений, которого вы хотите связать с организацией. Поставщик удостоверений может быть связан только с одной организацией.

Домен

Домен организации, который вы хотите связать с поставщиком удостоверений.

Скрыть на странице входа

Должен ли этот поставщик удостоверений быть скрыт на страницах входа, когда пользователь проходит аутентификацию в рамках организации.

Перенаправление при совпадении домена электронной почты

Должны ли участники автоматически перенаправляться к поставщику удостоверений, если их домен электронной почты совпадает с доменом, установленным для поставщика удостоверений.

После привязки к организации поставщиком удостоверений можно управлять так же, как и любым другим в области, перейдя в раздел «Поставщики удостоверений» в меню. Однако описанные здесь параметры доступны только при управлении поставщиком удостоверений в рамках организации. Единственным исключением является параметр «Скрыть на странице входа» , который представлен здесь для удобства.

Редактирование связанного поставщика удостоверений

Вы можете в любое время редактировать любые настройки организации связанного поставщика удостоверений. Процедура

1. В меню нажмите «Организации» и перейдите на вкладку «Поставщики удостоверений» .
2. Найдите поставщика удостоверений в списке.

   На этом этапе вы можете воспользоваться функцией поиска.
3. Нажмите кнопку действия (три точки) в конце строки.
4. Нажмите «Изменить» .
5. Внесите необходимые изменения.
6. Нажмите «Сохранить ».

Редактирование связанного поставщика удостоверений

Отключение поставщика удостоверений от организации

Когда поставщик удостоверений отсоединяется от организации, он остается доступным как поставщик уровня области, который больше не связан с организацией. Чтобы удалить отсоединенного поставщика, используйте раздел «Поставщики удостоверений» в меню.

Процедура

1. В меню нажмите «Организации» и перейдите на вкладку «Поставщики удостоверений» .
2. Найдите поставщика удостоверений в списке.

   На этом этапе вы можете воспользоваться возможностями поиска.
3. Нажмите кнопку действия (три точки) в конце строки.
4. Нажмите «Отключить поставщика» .

Отключение поставщика удостоверений

Аутентификация участников

Когда вы включаете организации для области, аутентификация пользователя изменяется. Если пользователь распознается как аутентифицирующийся в контексте организации, поток аутентификации изменяется на основе организации.

При создании области потоки аутентификации автоматически обновляются, чтобы включить определенные шаги для аутентификации и подключения членов организации. Обновленные потоки аутентификации:

- браузер
- первый вход брокера

Главное изменение в потоке браузера заключается в том, что по умолчанию используется вход с идентификацией, чтобы пользователи идентифицировались до запроса их учетных данных. Что касается потока входа первого брокера, главное изменение заключается в автоматическом добавлении пользователей в качестве членов организации после того, как они аутентифицируются через поставщика идентификации, связанного с организацией, и успешно завершают поток.

Выбор использования входа с идентификацией или нет зависит от наличия организации в области. Если организаций не существует, пользователь выполняет обычные шаги для аутентификации с использованием имени пользователя и пароля или любого другого шага, настроенного в потоке браузера. В противном случае у пользователя сначала запрашивается имя пользователя или адрес электронной почты для продолжения аутентификации в области.

Основная цель входа в систему с приоритетом идентификации — идентификация пользователя:

- Является ли пользователь существующим или новым?
- Является ли пользователь членом какой-либо организации в рамках сферы?
- Если пользователь является членом организации, связан ли он с каким-либо поставщиком удостоверений, связанным с этой организацией?

В зависимости от результата идентификации пользователя поток аутентификации изменяется: либо выполняется аутентификация с запросом учетных данных пользователя, либо пользователь автоматически перенаправляется для

аутентификации в рамках безопасности организации через поставщика удостоверений.

Понимание входа в систему с приоритетом идентификации

Помимо идентификации пользователя после предоставления имени, вход в систему с первичной идентификацией также отвечает за:

- Сопоставление домена электронной почты с организацией.
- Принятие решения о том, следует ли продолжать процесс аутентификации, если для указанного имени пользователя уже существует учетная запись
- Принятие решения о том, как должен проходить аутентификация пользователя, в зависимости от того, как настроены домены и поставщики удостоверений в организации.
- Простая аутентификация пользователей через поставщика удостоверений, связанного с организацией, если домен электронной почты совпадает с доменом, установленным для поставщика удостоверений

Вход с идентификацией предоставляет те же возможности, которые предоставляет обычная страница входа с полями имени пользователя и пароля. Пользователи по-прежнему могут самостоятельно зарегистрироваться, нажав на ссылку регистрации или выбрав любого брокера идентификации или социального брокера, который не связан с организацией в этой области.

Страница входа с указанием личности

В случае, если пользователь не существует, если он пытается пройти аутентификацию с использованием домена электронной почты, соответствующего домену организации, страница входа в систему с идентификацией появляется снова с сообщением о том, что предоставленное имя пользователя недействительно. На этом этапе нет необходимости спрашивать у пользователя учетные данные.

Идентификация — прежде всего, когда пользователь не существует

Существует несколько вариантов регистрации пользователя, позволяющих ему пройти аутентификацию в сфере и присоединиться к организации.

Если в области включена настройка самостоятельной регистрации, пользователь может нажать ссылку Register на странице входа identity-first и создать учетную запись в области. После этого администратор может отправить пользователю ссылку-приглашение или вручную добавить пользователя в качестве участника организации. Для получения более подробной информации см. Управление участниками .

Если в организации есть поставщик удостоверений без домена, а параметр Скрыть на странице входа выключен , пользователи также могут щелкнуть ссылку поставщика удостоверений на странице входа с приоритетом удостоверений, чтобы автоматически создать учетную запись и присоединиться к организации после аутентификации через поставщика удостоверений. Для получения более подробной информации см. Управление поставщиками удостоверений .

В ситуации, аналогичной предыдущей секции, организация может иметь поставщика удостоверений, установленного с одним из доменов организации. В этой ситуации пользователь перенаправляется к поставщику удостоверений, если адрес электронной почты этого пользователя совпадает с определенным доменом из организации. После завершения потока создается учетная запись, и пользователь присоединяется к организации.

Настройка существующих потоков аутентификации

Как уже упоминалось ранее для новых областей, потоки аутентификации автоматически обновляются необходимыми шагами для поддержки организаций и аутентификации их членов. Для существующих областей, в дополнение к включению организаций в область, вам также необходимо вручную обновить ваши существующие (пользовательские) потоки аутентификации.

Измените поток браузера , выполнив следующие действия:

Процедура

1. Дублируйте текущий поток, привязанный к типу привязки потока браузера, чтобы не нарушать поток, который вы используете в данный момент.

2. Нажмите «Добавить подпоток» и дайте ему имя, например «Моя организация».
3. Переместите недавно добавленный подпоток My Organization для выполнения сразу после шага выполнения Identity Provider Redirector . Главное здесь то, что подпоток должен произойти до любого другого подпотока или шага выполнения, который аутентифицирует пользователя с использованием любых учетных данных, которые вы поддерживаете в своей области. После добавления измените Requirement на Alternative .
4. Нажмите Добавить подпоток в подпотоке Моя организация и дайте ему имя, например Моя организация - Условный . После добавления измените Требование на Условный .
5. Нажмите Добавить условие в подпотоке Моя организация - Условное и выберите Условие - настроено пользователем . После добавления измените Требование на Обязательное .
6. Нажмите «Добавить шаг» в подпотоке « Моя организация — Условный» и выберите «*Идентификация организации — Первый вход».
    - Шаг выполнения. После добавления измените Требование на Альтернативу .
7. Свяжите поток аутентификации с типом привязки браузера .

Поток браузера организаций

После того как вы включите параметр «Организации» в области и создадите хотя бы одну организацию, вы сможете увидеть страницу входа с приоритетом идентификации и начать использовать организации в своей области.

Измените процесс входа в систему первого брокера , выполнив следующие действия:

Процедура

1. Дублируйте текущий поток, привязанный к типу привязки потока входа первого брокера, чтобы не нарушать поток, который вы используете в данный момент.

2. Нажмите Добавить подпоток и дайте ему имя, например Organization Member - Conditional. После добавления измените Требование на Условное .
3. Нажмите Добавить условие в подпотоке Член организации - Условный и выберите Условие - настроено пользователем . После добавления измените Требование на Обязательное .
4. Нажмите Add step в подпотоке Organization Member - Conditional и выберите шаг выполнения Organization Member Onboard . После добавления измените Requirement на Required .
5. Свяжите поток аутентификации с типом привязки входа первого брокера .

Организации первый брокер поток

Теперь вы сможете пройти аутентификацию с помощью любого поставщика удостоверений, связанного с организацией, и позволить пользователю присоединиться к организации в качестве участника сразу после завершения первого процесса входа в систему через браузер.

Картографирование претензий организаций

Чтобы сопоставить специфичные для организации утверждения в токены, клиенту необходимо запросить область организации при отправке запросов авторизации на сервер. При аутентификации в контексте организации клиенты могут запросить organizationобласть для сопоставления информации об организациях, членом которых является пользователь.

В результате токен будет содержать следующее утверждение:

```
"organization": {
  "testcorp": {
    "id": "42c3e46f-2477-44d7-a85b-d3b43f6b31fa",
    "attr1": [
      "value1"
    ]
  }
}
```

Утверждение организации может использоваться клиентами (например, из токенов ID) и серверами ресурсов (например, из токенов доступа) для авторизации доступа к защищенным ресурсам на основе организации, членом которой является пользователь.

Область organizationдействия — это встроенная необязательная клиентская область действия в области. Поэтому эта область действия добавляется к любому клиенту, созданному в области действия, по умолчанию. Она также определяет сопоставитель Organization Membership, который управляет тем, как информация о членстве в организации сопоставляется с токенами.

По умолчанию идентификатор организации и атрибуты не включены в заявку организации. Чтобы включить их, отредактируйте mapper и включите опции **Add organization id** и **Add organization attributes** соответственно.

Включение атрибутов в заявку организации

Объем organizationзапрашивается с использованием различных форматов:

Формат	Описание
`organization`	Сопоставляется с одной организацией, если пользователь является членом одной организации. В противном случае, если пользователь является членом нескольких организаций, пользователю будет предложено выбрать организацию при аутентификации в области.
`organization:<alias>`	Сопоставляется с одной организацией с указанным псевдонимом.
`organization:*`	Сопоставляет все организации, членом которых является пользователь.

Управление OpenID Connect и клиентами SAML

Клиенты — это сущности, которые могут запрашивать аутентификацию пользователя. Клиенты бывают двух видов. Первый тип клиента — это приложение, которое хочет участвовать в едином входе. Эти клиенты просто хотят, чтобы Tuxedo SSO обеспечивал им безопасность. Другой тип клиента — это тот, который запрашивает токен доступа, чтобы он мог вызывать другие службы от

имени аутентифицированного пользователя. В этом разделе обсуждаются различные аспекты настройки клиентов и различные способы сделать это.

Управление клиентами OpenID Connect

OpenID Connect — рекомендуемый протокол для защиты приложений. Он был разработан с нуля, чтобы быть дружелюбным к вебу, и лучше всего работает с приложениями HTML5/JavaScript.

Создание клиента OpenID Connect

Чтобы защитить приложение, использующее протокол OpenID Connect, вы создаете клиент.

Процедура

1. Нажмите «Клиенты» в меню.
2. Нажмите Создать клиента .

   Создать клиента
3. Оставьте тип клиента OpenID Connect .
4. Введите идентификатор клиента.

   Этот идентификатор представляет собой буквенно-цифровую строку, которая используется в запросах OIDC и в базе данных Tuxedo SSO для идентификации клиента.
5. Укажите имя клиента.

   Если вы планируете локализовать это имя, настройте заменяющее строковое значение. Например, строковое значение, такое как ${myapp}. Для получения дополнительной информации см. Руководство разработчика сервера .
6. Нажмите «Сохранить ».

Это действие создаст клиент и перенаправит вас на вкладку «Настройки» , где вы сможете выполнить базовую настройку .

Базовая конфигурация

Вкладка «Настройки» содержит множество параметров для настройки этого клиента.

Вкладка «Настройки»

Общие настройки

Идентификатор клиента

Буквенно-цифровая строка идентификатора, которая используется в запросах OIDC и в базе данных Tuxedo SSO для идентификации клиента.

Имя

Имя клиента на экране пользовательского интерфейса Tuxedo SSO. Чтобы локализовать имя, задайте заменяющее строковое значение. Например, строковое значение, такое как ${myapp}. Для получения дополнительной информации см. Руководство разработчика сервера .

Описание

Описание клиента. Этот параметр также может быть локализован.

Всегда отображать в консоли

Всегда указывать этого клиента в консоли учетной записи, даже если у этого пользователя нет активного сеанса.

Настройки доступа

Корневой URL-адрес

Если Tuxedo SSO использует какие-либо настроенные относительные URL-адреса, это значение добавляется к ним.

Домашняя страница URL

Предоставляет URL-адрес по умолчанию, когда серверу аутентификации необходимо выполнить перенаправление или обратную ссылку на клиента.

Допустимые URI перенаправления

Обязательное поле. Введите шаблон URL и нажмите + для добавления и - для удаления существующих URL и нажмите Сохранить . Точное (с учетом регистра) совпадение строк используется для сравнения допустимых URI перенаправления.

Вы можете использовать подстановочные знаки в конце шаблона URL. Например http://host.com/path/*, . Чтобы избежать проблем безопасности, если переданный URI перенаправления содержит часть userinfo или его путь управляет доступом к родительскому каталогу (/../), сравнение с подстановочными знаками не выполняется, а выполняется стандартное и безопасное точное сопоставление строк.

Полный подстановочный *действительный URI перенаправления также может быть настроен для разрешения любого URI перенаправления http или https . Пожалуйста, не используйте его в производственных средах.

Эксклюзивные шаблоны URI перенаправления обычно более безопасны. См. Unspecific Redirect URIs для получения дополнительной информации.

Веб-источники

Введите шаблон URL и нажмите +, чтобы добавить и -, чтобы удалить существующие URL. Нажмите Сохранить.

Эта опция обрабатывает Cross-Origin Resource Sharing (CORS) . Если браузер JavaScript пытается выполнить AJAX HTTP-запрос к серверу, домен которого отличается от того, с которого пришел код JavaScript, запрос должен использовать CORS. Сервер должен обрабатывать запросы CORS, в противном случае браузер не отобразит или не разрешит обработку запроса. Этот протокол защищает от XSS, CSRF и других атак на основе JavaScript.

Перечисленные здесь URL-адреса доменов встроены в токен доступа, отправленный клиентскому приложению. Клиентское приложение использует эту информацию для принятия решения о том, разрешить ли ему вызывать запрос CORS. Эту функцию поддерживают только клиентские адаптеры Tuxedo SSO. Для получения дополнительной информации см. Руководства по защите приложений .

URL-адрес администратора

Конечная точка обратного вызова для клиента. Сервер использует этот URL для выполнения обратных вызовов, таких как отправка политик отзыва, выполнение выхода из обратного канала и другие административные операции. Для адаптеров сервлета Tuxedo SSO этот URL может быть корневым URL приложения сервлета. Для получения дополнительной информации см. Руководства по защите приложений .

Конфигурация возможностей

Аутентификация клиента

Тип клиента OIDC.

- НА

  Для серверных клиентов, которые выполняют вход в браузер и требуют клиентские секреты при выполнении запроса токена доступа. Этот параметр следует использовать для серверных приложений.

- ВЫКЛЮЧЕННЫЙ

  Для клиентов на стороне клиента, которые выполняют вход в браузер. Поскольку невозможно гарантировать, что секреты будут храниться в безопасности с помощью клиентов на стороне клиента, важно ограничить доступ, настроив правильные URI перенаправления.

Авторизация

Включает или отключает поддержку детальной авторизации для этого клиента.

Стандартный поток

Если этот параметр включен, этот клиент может использовать поток кода авторизации OIDC .

Гранты прямого доступа

Если этот параметр включен, этот клиент может использовать гранты прямого доступа OIDC .

Неявный поток

Если этот параметр включен, этот клиент может использовать неявный поток OIDC .

Роли учетной записи службы

Если включено, этот клиент может пройти аутентификацию в Tuxedo SSO и получить токен доступа, выделенный для этого клиента. В терминах спецификации OAuth2 это включает поддержку для Client Credentials Grantэтого клиента.

Разрешение на авторизацию устройства Auth 2.0

Если включено, этот клиент может использовать грант авторизации устройства OIDC .

Грант OIDC CIBA

Если включено, этот клиент может использовать грант аутентификации обратного канала, инициированный клиентом OIDC .

Настройки входа

Тема входа

Тема для использования на страницах входа в систему, одноразовых паролей, предоставления регистрации и восстановления забытого пароля.

Требуется согласие

Если эта функция включена, пользователи должны дать согласие на клиентский доступ.

Для клиентов на стороне клиента, которые выполняют вход в браузер. Поскольку невозможно гарантировать, что секреты будут храниться в безопасности с помощью клиентов на стороне клиента, важно ограничить доступ, настроив правильные URI перенаправления.

Отображение клиента на экране

Этот переключатель применяется, если параметр «Требуется согласие» отключен .

- Выключенный

  Экран согласия будет содержать только согласия, соответствующие настроенным клиентским областям действия.

- На

  На экране согласия также будет один пункт, касающийся самого клиента.

Текст экрана согласия клиента

Применяется, если включены параметры Consent required и Display client on screen . Содержит текст, который будет отображаться на экране согласия о разрешениях для этого клиента.

Настройки выхода

Выход из переднего канала

Если включен выход из системы Front Channel , приложение должно иметь возможность выходить из системы пользователей через фронтальный канал в соответствии со спецификацией OpenID Connect Front-Channel Logout . Если включено, вы также должны предоставить Front-Channel Logout URL.

URL выхода из переднего канала

URL-адрес, который будет использоваться Tuxedo SSO для отправки клиентам запросов на выход из системы через фронтальный канал.

URL выхода из обратного канала

URL, который заставит клиента выйти из системы, когда запрос на выход отправляется в эту область (через end_session_endpoint). Если этот параметр пропущен, клиенту не отправляются запросы на выход.

Требуется выход из сеанса обратного канала

Указывает, включается ли утверждение идентификатора сеанса в токен выхода из системы при использовании URL-адреса выхода из системы Backchannel .

Выход из обратного канала отменяет автономные сеансы

Указывает, включено ли событие revoke_offline_access в токен выхода из системы при использовании URL-адреса выхода из системы Backchannel. Tuxedo SSO отменит офлайн-сессии при получении токена выхода из системы с этим событием.

Расширенная конфигурация

После заполнения полей на вкладке Настройки вы можете использовать другие вкладки для выполнения расширенной настройки. Например, вы можете использовать вкладки Разрешения и Роли для настройки детальной аутентификации для администраторов. См. Детальные разрешения администратора . Также см. оставшиеся разделы в этой главе для других возможностей.

Вкладка «Дополнительно»
При нажатии на вкладку «Дополнительно» отображаются дополнительные поля. Для получения подробной информации о конкретном поле щелкните значок вопросительного знака для этого поля. Однако некоторые поля подробно описаны в этом разделе.

Детальная конфигурация OpenID Connect
URL-адрес логотипа

URL-адрес, ссылающийся на логотип клиентского приложения.

URL-адрес политики

URL-адрес, который Клиент проверяющей стороны предоставляет Конечному пользователю для прочтения информации о том, как будут использоваться данные профиля.

URL-адрес условий обслуживания

URL-адрес, который Клиент Проверяющей стороны предоставляет Конечному пользователю для ознакомления с условиями обслуживания Проверяющей стороны.

Поддержка подписанных и зашифрованных идентификаторов токенов

Tuxedo SSO может шифровать токены ID в соответствии со спецификацией Json Web Encryption (JWE) . Администратор определяет, шифруются ли токены ID для каждого клиента.

Ключ, используемый для шифрования токена ID, — это ключ шифрования контента (CEK). Tuxedo SSO и клиент должны договориться, какой CEK использовать и как он доставляется. Метод, используемый для определения CEK, — это режим управления ключами. Режим управления ключами, поддерживаемый Tuxedo SSO, — это шифрование ключей.

В ключевом шифровании:

1. Клиент генерирует асимметричную пару криптографических ключей.
2. Открытый ключ используется для шифрования CEK.
3. Tuxedo SSO генерирует CEK для каждого токена идентификатора
4. Tuxedo SSO шифрует идентификационный токен, используя сгенерированный CEK
5. Tuxedo SSO шифрует CEK, используя открытый ключ клиента.

6. Клиент расшифровывает этот зашифрованный CEK, используя свой закрытый ключ.
7. Клиент расшифровывает идентификационный токен, используя расшифрованный CEK.

Ни одна сторона, кроме клиента, не может расшифровать идентификационный токен.

Клиент должен передать свой открытый ключ для шифрования CEK в Tuxedo SSO. Tuxedo SSO поддерживает загрузку открытых ключей с URL, предоставленного клиентом. Клиент должен предоставить открытые ключи в соответствии со спецификацией Json Web Keys (JWK) .

Процедура следующая:

1. Откройте вкладку «Ключи » клиента .
2. Переключите URL-адрес JWKS в положение ВКЛ.
3. Введите URL открытого ключа клиента в текстовое поле URL JWKS .

Алгоритмы Key Encryption определены в спецификации Json Web Algorithm (JWA) . Tuxedo SSO поддерживает:

- RSAES-PKCS1-v1_5(RSA1_5)
- RSAES OAEP с использованием параметров по умолчанию (RSA-OAEP)
- RSAES OAEP 256 с использованием SHA-256 и MFG1 (RSA-OAEP-256)

Процедура выбора алгоритма следующая:

1. Откройте вкладку «Дополнительно» клиента .
2. Открытая детальная конфигурация OpenID Connect .
3. Выберите алгоритм из раскрывающегося меню «Алгоритм шифрования содержимого идентификатора токена» .

Режимы совместимости OpenID Connect
Этот раздел существует для обратной совместимости. Нажмите на значки с вопросительным знаком для получения подробной информации по каждому полю.

Включены токены доступа, привязанные к сертификату OAuth 2.0 Mutual TLS

Mutual TLS связывает токен доступа и токен обновления вместе с клиентским сертификатом, который обменивается во время рукопожатия TLS. Эта привязка не позволяет злоумышленнику использовать украденные токены.

Этот тип токена — токен-держатель ключа. В отличие от токенов-предъявителей, получатель токена-держателя ключа может проверить, является ли отправитель токена законным.

Если этот параметр включен, рабочий процесс будет следующим:

1. Запрос токена отправляется на конечную точку токена в потоке кода авторизации или гибридном потоке.
2. Tuxedo SSO запрашивает клиентский сертификат.
3. Tuxedo SSO получает клиентский сертификат.
4. Tuxedo SSO успешно проверяет клиентский сертификат.

Если проверка не пройдена, Tuxedo SSO отклоняет токен.

В следующих случаях Tuxedo SSO проверит клиента, отправляющего токен доступа или токен обновления:

- Запрос на обновление токена отправляется на конечную точку токена с токеном обновления держателя ключа.
- Запрос UserInfo отправляется на конечную точку UserInfo с токеном доступа владельца ключа.
- Запрос на выход из системы отправляется на конечную точку выхода Tuxedo SSO, не соответствующую OIDC, с токеном обновления владельца ключа.

Более подробную информацию см . в разделе Взаимная аутентификация клиента TLS и токены доступа, привязанные к сертификату OAuth 2.0.

> Клиентские адаптеры Tuxedo SSO не поддерживают проверку токена владельца ключа. Адаптеры Tuxedo SSO обрабатывают токены доступа и обновления как токены на предъявителя.

OAuth 2.0 демонстрирует доказательство владения на уровне приложений (DPoP)

DPoP связывает токен доступа и токен обновления с публичной частью пары ключей клиента. Эта привязка не позволяет злоумышленнику использовать украденные токены.

Этот тип токена — токен-держатель ключа. В отличие от токенов-предъявителей, получатель токена-держателя ключа может проверить, является ли отправитель токена законным.

Если переключатель клиента OAuth 2.0 DPoP Bound Access Tokens Enabledвключен, рабочий процесс следующий:

1. Запрос токена отправляется на конечную точку токена в потоке кода авторизации или гибридном потоке.
2. Tuxedo SSO запрашивает доказательство DPoP.
3. Tuxedo SSO получает доказательство DPoP.
4. Tuxedo SSO успешно проверяет доказательство DPoP.

Если проверка не пройдена, Tuxedo SSO отклоняет токен.

Если переключатель OAuth 2.0 DPoP Bound Access Tokens Enabledвыключен, клиент все равно может отправить DPoPдоказательство в запросе токена. В этом случае Tuxedo SSO проверит доказательство DPoP и добавит отпечаток к токену. Но если переключатель выключен, привязка DPoP не применяется сервером Tuxedo SSO для этого клиента. Рекомендуется включить этот переключатель, если вы хотите убедиться, что конкретный клиент всегда использует привязку DPoP.

В следующих случаях Tuxedo SSO проверит клиента, отправляющего токен доступа или токен обновления:

- Запрос на обновление токена отправляется на конечную точку токена с токеном обновления держателя ключа. Эта проверка выполняется только для публичных клиентов, как описано в спецификации DPoP. Для конфиденциальных клиентов проверка не выполняется, поскольку выполняется аутентификация клиента с надлежащими учетными данными клиента, чтобы гарантировать, что запрос исходит от законного клиента. Для публичных клиентов как токены доступа, так и токены обновления

привязаны к DPoP. Для конфиденциальных клиентов привязаны только токены доступа.

- Запрос UserInfo отправляется на конечную точку UserInfo с токеном доступа владельца ключа.
- Запрос на выход отправляется на несовместимую с OIDC фирменную конечную точку выхода Tuxedo SSO Конечная точка выхода с токеном обновления держателя ключа. Эта проверка выполняется только для публичных клиентов, как описано выше.

Более подробную информацию см . в разделе «Подтверждение права собственности OAuth 2.0 (DPoP)» .

Адаптеры клиента Tuxedo SSO не поддерживают проверку токена держателя ключа DPoP. Адаптеры Tuxedo SSO обрабатывают токены доступа и обновления как токены носителя.

DPoP — это **предварительная версия** , которая не поддерживается полностью. Эта функция отключена по умолчанию.

Чтобы включить запустите сервер с помощью `--features=preview`или`--features=dpop`

Расширенные настройки для OIDC

Расширенные настройки OpenID Connect позволяют настраивать переопределения на уровне клиента для тайм-аутов сеанса и токена .

Конфигурация	Описание
Срок действия токена доступа	Значение переопределяет параметр области с тем же именем.
Клиентский сеанс в режиме ожидания	Значение переопределяет параметр области с тем же именем. Значение должно быть короче, чем глобальный **SSO Session Idle** .
Макс. сеанс клиента	Значение переопределяет параметр области с тем же именем. Значение должно быть короче, чем глобальный **SSO Session Max** .
Клиентский автономный сеанс в режиме ожидания	Этот параметр позволяет настроить более короткий тайм-аут простоя офлайн-сеанса для клиента. Тайм-аут — это количество времени, в течение которого сеанс остается бездействующим, прежде чем Tuxedo SSO отзовет свой офлайн-токен. Если не задано, используется

Конфигурация	Описание
	область Offline Session Idle .
Максимальный офлайн-сеанс клиента	Этот параметр позволяет вам настроить более короткий максимальный срок жизни офлайн-сессии для клиента. Срок жизни — это максимальное время, прежде чем Tuxedo SSO отзовет соответствующий офлайн-токен. Для этой опции необходимо включить Offline Session Max Limited глобально в области, и по умолчанию установлено значение Offline Session Max .

Ключ доказательства для метода проверки кода обмена кодами

Если злоумышленник украдет код авторизации законного клиента, Proof Key for Code Exchange (PKCE) не позволит злоумышленнику получить токены, которые применяются к коду.

Администратор может выбрать один из следующих вариантов:

(пустой)

> Tuxedo SSO не применяет PKCE, если клиент не отправит соответствующие параметры PKCE в конечную точку авторизации Tuxedo SSOs.

C256

> Tuxedo SSO применяется к клиентскому PKCE, метод проверки кода которого — S256.

простой

> Tuxedo SSO применяется к клиентскому PKCE, метод проверки кода которого является простым.

Более подробную информацию см. в документе RFC 7636 «Ключ подтверждения для обмена кодами публичными клиентами OAuth» .

Сопоставление ACR с уровнем аутентификации (LoA)

В расширенных настройках клиента вы можете определить, какое Authentication Context Class Reference (ACR)значение сопоставляется с каким Level of Authentication (LoA). Это сопоставление может быть указано также в области, как

указано в Сопоставлении ACR с LoA . Лучшей практикой является настройка этого сопоставления на уровне области, что позволяет совместно использовать одни и те же настройки для нескольких клиентов.

Может Default ACR Valuesиспользоваться для указания значений по умолчанию, когда запрос на вход отправляется от этого клиента в Tuxedo SSO без acr_valuesпараметра и без claimsпараметра, имеющего acrприкрепленное утверждение. См. официальную спецификацию динамической регистрации клиента OIDC .

> Обратите внимание, что значения ACR по умолчанию используются в качестве уровня по умолчанию, однако их нельзя надежно использовать для принудительного входа с определенным уровнем. Например, предположим, что вы настраиваете `Default ACR Values`на уровень 2. Тогда по умолчанию пользователи должны будут проходить аутентификацию с уровнем 2. Однако, когда пользователь явно прикрепляет параметр к запросу на вход, например `acr_values=1`, то будет использоваться уровень 1. В результате, если клиенту действительно требуется уровень 2, клиенту рекомендуется проверить наличие утверждения `acr`внутри токена ID и дважды проверить, что он содержит запрошенный уровень 2.

Более подробную информацию см. в разделе «Пошаговая аутентификация» и официальной спецификации OIDC .

Конфиденциальные данные клиента

Если для параметра «Аутентификация клиента» установлено значение «ВКЛ» , учетные данные клиента необходимо настроить на вкладке «Учетные данные» .

Вкладка «Учетные данные»

Раскрывающийся список «Аутентификатор клиента» определяет тип учетных данных, которые следует использовать для вашего клиента .

Идентификатор клиента и секрет

Этот выбор является настройкой по умолчанию. Секрет генерируется автоматически. Нажмите Regenerate, чтобы повторно создать секрет, если необходимо.

Подписанный JWT

Подписанный JWT — это «подписанный Json Web Token».

При выборе этого типа учетных данных вам также придется сгенерировать закрытый ключ и сертификат для клиента на вкладке Keys. Закрытый ключ будет использоваться для подписи JWT, в то время как сертификат будет использоваться сервером для проверки подписи.

Вкладка «Ключи»

Нажмите на Generate new keysкнопку, чтобы начать этот процесс.

Генерировать ключи

1. Выберите формат архива, который вы хотите использовать.
2. Введите ключевой пароль .
3. Введите пароль магазина .
4. Нажмите «Создать» .

Когда вы генерируете ключи, Tuxedo SSO сохраняет сертификат, а вы загружаете закрытый ключ и сертификат для своего клиента.

Вы также можете сгенерировать ключи с помощью внешнего инструмента, а затем импортировать сертификат клиента, нажав кнопку Импорт сертификата .

Импортный сертификат

1. Выберите формат архива сертификата.
2. Введите пароль магазина.
3. Выберите файл сертификата, нажав кнопку Импорт файла .
4. Нажмите Импорт .

Импорт сертификата не нужен, если вы нажмете Use JWKS URL . В этом случае вы можете указать URL, где публикуется открытый ключ в формате JWK . С этой опцией, если ключ когда-либо изменится, Tuxedo SSO повторно импортирует ключ.

Если вы используете клиент, защищенный адаптером Tuxedo SSO, вы можете настроить URL-адрес JWKS в следующем формате, предполагая,

что https://myhost.com/myapp является корневым URL-адресом вашего клиентского приложения:

https://myhost.com/myapp/k_jwks

Более подробную информацию см. в Руководстве разработчика сервера .

Подписанный JWT с клиентским секретом

Если вы выберете эту опцию, вы сможете использовать JWT, подписанный секретом клиента, вместо закрытого ключа.

Секрет клиента будет использоваться для подписи JWT клиентом.

Сертификат X509

Tuxedo SSO проверит, использует ли клиент правильный сертификат X509 во время установления связи TLS.

сертификат X509

Валидатор также проверяет поле Subject DN сертификата с настроенным выражением проверки regexp. Для некоторых случаев использования достаточно принять все сертификаты. В этом случае можно использовать (.*?)(?:$)выражение.

Для Tuxedo SSO существуют два способа получения идентификатора клиента из запроса:

- Параметр client_idв запросе (описанный в разделе 2.2 спецификации OAuth 2.0).
- Поставка client_idкак параметр формы.

Секретная ротация клиента

> Обратите внимание, что поддержка Client Secret Rotation находится в стадии разработки. Используйте эту функцию экспериментально.

Для клиента с аутентификацией «Конфиденциальный клиент» Tuxedo SSO поддерживает функцию ротации клиентских секретов с помощью клиентских политик .

Политика ротации секретов клиента обеспечивает большую безопасность для устранения таких проблем, как утечка секретов. После включения Tuxedo SSO поддерживает до двух одновременно активных секретов для каждого клиента. Политика управляет ротациями в соответствии со следующими настройками:

- Срок действия секрета: [секунды] — при ротации секрета это время истечения срока действия нового секрета. Количество секунд , добавляемое к дате создания секрета. Рассчитывается во время выполнения политики.
- Срок действия ротированного секрета: [секунды] — при ротации секрета это значение представляет собой оставшееся время действия старого секрета. Это значение всегда должно быть меньше срока действия секрета. Если значение равно 0, старый секрет будет немедленно удален во время ротации клиента. Количество секунд , добавляемое к дате ротации секрета. Рассчитывается во время выполнения политики.
- Оставшееся время истечения срока действия для ротации во время обновления: [секунды] — период времени, когда обновление динамического клиента должно выполнить ротацию секрета клиента. Рассчитывается во время выполнения политики.

При ротации клиентского секрета генерируется новый основной секрет, а старый клиентский основной секрет становится вторичным секретом с новой датой истечения срока действия.

Правила ротации секретной информации клиентов
Ротации не происходят автоматически или через фоновый процесс. Для выполнения ротации требуется действие обновления на клиенте, либо через консоль администратора Tuxedo SSO с помощью функции Regenerate Secret на вкладке учетных данных клиента, либо через API REST администратора. При вызове действия обновления клиента ротация секрета происходит в соответствии с правилами:

- Когда значение срока действия секрета меньше текущей даты.
- Во время динамической регистрации клиента при запросе на обновление клиентского секрета клиент будет автоматически ротирован, если

значение оставшегося срока действия для ротации во время обновления совпадает с периодом между текущей датой и истечением срока действия секрета .

Кроме того, с помощью Admin REST API можно в любое время принудительно выполнить ротацию клиентского секрета.

При создании новых клиентов, если активна политика ротации секретов клиентов, поведение будет применено автоматически.

Чтобы применить поведение ротации секретов к существующему клиенту, обновите этот клиент после определения политики, чтобы применить это поведение.

Создание политики ротации секретов клиента OIDC

Ниже приведен пример определения политики секретной ротации:

Процедура

1. Нажмите «Настройки области» в меню.
2. Перейдите на вкладку «Политики клиента» .
3. На странице «Профили» нажмите «Создать профиль клиента» .

   Создать профиль
4. Введите любое имя в поле Имя .
5. Введите описание, которое поможет вам определить цель профиля для Описание .
6. Нажмите «Сохранить ».

   Это действие создает профиль и позволяет вам настраивать исполнителей.
7. Нажмите Добавить исполнителя , чтобы настроить исполнителя для этого профиля.

   Создать профиль исполнителя
8. Выберите секретную ротацию для типа исполнителя .

9. Введите максимальное время действия каждого секрета в секундах для параметра Срок действия секрета .
10. Введите максимальное время действия каждого ротируемого секрета в секундах для параметра « Истечение срока действия ротируемого секрета» .

    Помните, что значение **«Срок действия ротированного секрета»** всегда должно быть меньше « **Срока действия секрета»** .
11. Введите количество времени в секундах, по истечении которого любое действие обновления обновит клиент для оставшегося времени истечения срока действия .
12. Нажмите Добавить .

    В примере выше:

    - Каждый секрет действителен в течение одной недели.
    - Срок действия ротируемого секрета истекает через два дня.
    - Окно обновления динамических клиентов начинается за один день до истечения срока действия секрета.
13. Вернитесь на вкладку «Политики клиента» .
14. Нажмите Политики .
15. Нажмите Создать клиентскую политику .

    Создайте политику ротации клиентских секретов
16. Введите любое имя в поле Имя .
17. Введите описание, которое поможет вам определить цель политики для Описание .
18. Нажмите «Сохранить ».

    Это действие создает политику и позволяет вам связывать политики с профилями. Оно также позволяет вам настраивать условия для выполнения политики.
19. В разделе «Условия» нажмите «Добавить условие» .

Создайте условие политики ротации секретных данных клиента

20. Чтобы применить поведение ко всем конфиденциальным клиентам, выберите client-access-type в поле Condition Type .

    Чтобы применить к определенной группе клиентов, другой подход — выбрать тип *клиентских ролей* в поле **Тип условия** . Таким образом, вы можете создать определенные роли и назначить пользовательскую конфигурацию ротации для каждой роли.

21. Добавьте конфиденциальность в поле Тип доступа клиента .
22. Нажмите Добавить .
23. Вернитесь к настройкам политики, в разделе Профили клиентов нажмите Добавить профиль клиента , затем выберите Профиль ротации еженедельных секретов клиента из списка и нажмите Добавить .

Политика ротации клиентских секретов

Чтобы применить секретное поведение ротации к существующему клиенту, выполните следующие действия:

Использование консоли администратора

1. Нажмите **«Клиенты»** в меню.
2. Нажмите на клиента.
3. Перейдите на вкладку **«Учетные данные»** .
4. Нажмите **«Повторно сгенерировать** секрет клиента».

Используя клиентские REST-сервисы, это можно выполнить двумя способами:

- Через операцию обновления на клиенте
- Через конечную точку секретного клиента для повторной генерации

Использование учетной записи службы

Каждый клиент OIDC имеет встроенную учетную запись службы . Используйте эту учетную запись службы для получения токена доступа.

Процедура

1. Нажмите «Клиенты» в меню.
2. Выберите своего клиента.
3. Откройте вкладку «Настройки» .
4. Включите аутентификацию клиента .
5. Выберите роли учетных записей служб .
6. Нажмите «Сохранить ».
7. Настройте учетные данные клиента .
8. Перейдите на вкладку «Область действия» .
9. Убедитесь, что у вас есть роли, или переключите параметр «Полная область действия разрешена» в положение «ВКЛ» .
10. Нажмите вкладку «Роли учетной записи службы» .
11. Настройте роли, доступные для этой учетной записи службы для вашего клиента.

Роли токенов доступа являются пересечением:

- Сопоставления областей ролей клиента объединены с сопоставлениями областей ролей, унаследованными от связанных областей клиентов.
- Роли учетных записей служб.

URL-адрес REST для вызова — /realms/{realm-name}/protocol/openid-connect/token. Этот URL-адрес должен быть вызван как запрос POST и требует, чтобы вы отправили учетные данные клиента вместе с запросом.

По умолчанию учетные данные клиента представлены clientId и clientSecret клиента в заголовке Authorization: Basic , но вы также можете аутентифицировать клиента с помощью подписанного утверждения JWT или любого другого настраиваемого механизма аутентификации клиента.

Вам также необходимо установить параметр grant_type на «client_credentials» в соответствии со спецификацией OAuth2.

Например, вызов POST для получения учетной записи службы может выглядеть следующим образом:

```
POST /realms/demo/protocol/openid-connect/token
Authorization: Basic cHJvZHVjdC1zYS1jbGllbnQ6cGFzc3dvcmQ=
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
```

Ответ будет похож на этот ответ токена доступа из спецификации OAuth 2.0.

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
    "access_token":"2YotnFZFEjr1zCsicMWpAA",
    "token_type":"bearer",
    "expires_in":60
}
```

По умолчанию возвращается только токен доступа. По умолчанию токен обновления не возвращается, и сеанс пользователя не создается на стороне Tuxedo SSO при успешной аутентификации. Из-за отсутствия токена обновления требуется повторная аутентификация, когда истекает срок действия токена доступа. Однако эта ситуация не означает дополнительных накладных расходов для сервера Tuxedo SSO, поскольку сеансы по умолчанию не создаются.

В этой ситуации выход из системы не нужен. Однако выданные токены доступа можно отозвать, отправив запросы на конечную точку отзыва OAuth2, как описано в разделе Конечные точки OpenID Connect .

Дополнительные ресурсы

Более подробную информацию см. в разделе « Предоставление учетных данных клиента» .

Поддержка аудитории

Обычно среда, в которой развернут Tuxedo SSO, состоит из набора конфиденциальных или публичных клиентских приложений, использующих Tuxedo SSO для аутентификации.

Также доступны службы (серверы ресурсов в спецификации OAuth 2), которые обслуживают запросы клиентских приложений и предоставляют ресурсы этим приложениям. Эти службы требуют отправки им токена доступа (токена носителя) для аутентификации запроса. Этот токен получается приложением frontend при входе в Tuxedo SSO.

В среде, где доверие между службами низкое, вы можете столкнуться со следующим сценарием:

1. Клиентское приложение внешнего интерфейса требует аутентификации с помощью Tuxedo SSO.
2. Tuxedo SSO аутентифицирует пользователя.
3. Tuxedo SSO выдает токен приложению.
4. Приложение использует токен для вызова недоверенной службы.
5. Недоверенная служба возвращает ответ приложению. Однако она сохраняет токен приложения.
6. Затем ненадежная служба вызывает доверенную службу, используя токен приложения. Это приводит к нарушению безопасности, поскольку ненадежная служба использует токен для доступа к другим службам от имени клиентского приложения.

Этот сценарий маловероятен в средах с высоким уровнем доверия между службами, но не в средах с низким уровнем доверия. В некоторых средах этот рабочий процесс может быть правильным, поскольку ненадежной службе может потребоваться извлечь данные из доверенной службы, чтобы вернуть данные исходному клиентскому приложению.

Неограниченная аудитория полезна, когда между службами существует высокий уровень доверия. В противном случае аудитория должна быть ограничена. Вы можете ограничить аудиторию и в то же время разрешить ненадежным службам извлекать данные из доверенных служб. В этом случае убедитесь, что ненадежная служба и доверенная служба добавлены в токен в качестве аудиторий.

Чтобы предотвратить любое нецелевое использование токена доступа, ограничьте аудиторию токена и настройте свои службы для проверки аудитории токена. Поток изменится следующим образом:

1. Приложение внешнего интерфейса проходит аутентификацию с помощью Tuxedo SSO.
2. Tuxedo SSO аутентифицирует пользователя.
3. Tuxedo SSO выдает токен приложению. Приложение знает, что ему нужно будет вызвать ненадежную службу, поэтому оно помещает scope=<untrusted service> в запрос аутентификации, отправленный в Tuxedo SSO (см. раздел Client Scopes для получения более подробной информации о параметре scope).

   Токен, выданный приложению, содержит ссылку на недоверенную службу в своей аудитории ("audience": ["<untrusted service>"]), которая объявляет, что клиент использует этот токен доступа для вызова недоверенной службы.
4. Недоверенная служба вызывает доверенную службу с помощью токена. Вызов не удался, поскольку доверенная служба проверяет аудиторию на токене и обнаруживает, что ее аудитория предназначена только для недоверенной службы. Такое поведение ожидаемо, и безопасность не нарушена.

Если клиент захочет вызвать доверенную службу позже, он должен получить другой токен, повторно выпустив логин SSO с scope=<trusted service> . Возвращенный токен будет содержать доверенную службу в качестве аудитории:

"audience": ["<trusted service>"]

Используйте это значение для вызова <доверенной службы> .

Настраивать
При настройке проверки аудитории:

- Убедитесь, что службы настроены на проверку аудитории по отправленному им токену доступа. Это может быть сделано способом, специфичным для вашего клиентского адаптера OIDC, который вы используете для защиты своего клиентского приложения OIDC.
- Убедитесь, что токены доступа, выданные Tuxedo SSO, содержат все необходимые аудитории. Аудитории можно добавлять с помощью ролей клиента, как описано в следующем разделе , или жестко закодировать. См. Жестко закодированная аудитория .

Автоматически добавлять аудиторию
Сопоставитель протокола Audience Resolve определяется в ролях области клиента по умолчанию . Сопоставитель проверяет клиентов, у которых есть хотя бы одна доступная роль клиента для текущего токена. Затем идентификатор клиента каждого клиента добавляется в качестве аудитории, что полезно, если ваши клиенты службы полагаются на роли клиента. Клиент службы обычно может быть клиентом без включенных потоков, который может не иметь никаких токенов, выпущенных непосредственно для себя. Он представляет собой сервер ресурсов OAuth 2 .

Например, для клиента службы и конфиденциального клиента вы можете использовать токен доступа, выданный для конфиденциального клиента, для вызова службы REST клиента службы. Клиент службы будет автоматически добавлен в качестве аудитории к токену доступа, выданному для конфиденциального клиента, если выполняются следующие условия:

- Клиент сервиса имеет все клиентские роли, определенные для него самого.
- Целевому пользователю назначена как минимум одна из этих клиентских ролей.
- Конфиденциальный клиент имеет сопоставления областей действия ролей для назначенной роли.

Если вы хотите убедиться, что аудитория не добавляется автоматически, не настраивайте

сопоставления областей ролей непосредственно на конфиденциальном клиенте. Вместо этого вы можете создать выделенную клиентскую область, которая содержит сопоставления областей ролей для клиентских ролей вашей выделенной клиентской области.

Предполагая, что область клиента добавлена как необязательная область клиента к конфиденциальному клиенту, роли клиента и аудитория будут добавлены в токен, если это явно запрошено параметром **scope=<trusted service>** .

Сам клиент интерфейса не добавляется автоматически в аудиторию токена доступа, что позволяет легко различать токен доступа и токен идентификатора, поскольку токен доступа не будет содержать клиента, для которого выдан токен в качестве аудитории.

Если вам нужен сам клиент в качестве аудитории, см. опцию жестко закодированной аудитории . Однако использование одного и того же клиента в качестве фронтенда и REST-сервиса не рекомендуется.

Жестко заданная аудитория

Если ваша служба полагается на роли области или вообще не полагается на роли в токене, может быть полезно использовать жестко закодированную аудиторию. Жестко закодированная аудитория — это сопоставитель протоколов, который добавит идентификатор клиента указанного клиента службы в качестве аудитории в токен. Вы можете использовать любое пользовательское значение, например URL, если вы хотите использовать аудиторию, отличную от идентификатора клиента.

Вы можете добавить протокол-картограф напрямую в клиент frontend. Если протокол-картограф добавляется напрямую, аудитория всегда будет также добавлена.

Для большего контроля над сопоставителем протоколов вы можете создать сопоставитель протоколов в выделенной клиентской области, которая будет называться, например, good-service .

Картограф протокола аудитории

- На вкладке Client details клиента good-service вы можете сгенерировать конфигурацию адаптера и подтвердить, что verify-token-audience установлен на true . Это действие заставляет адаптер проверять аудиторию, если вы используете эту конфигурацию.

- Вам необходимо убедиться, что конфиденциальный клиент может запросить хорошее обслуживание в качестве аудитории в своих токенах.

  О конфиденциальном клиенте:

  1. Перейдите на вкладку «Области действия клиента» .
  2. Назначьте хорошее обслуживание в качестве необязательной (или стандартной) клиентской области.

     Более подробную информацию см. в разделе «Связывание клиентских областей» .

- При желании вы можете оценить клиентские области и сгенерировать пример токена доступа. good-service будет добавлен к аудитории сгенерированного токена доступа, если good-service включен в параметр области действия , когда вы назначили его как необязательную клиентскую область действия.
- В вашем конфиденциальном клиентском приложении убедитесь, что используется параметр scope . Значение good-service должно быть включено, когда вы хотите выдать токен для доступа к good-service .

  Видеть:

  1. Адаптер JavaScript Tuxedo SSO в разделе «Защита приложений» , если ваше приложение использует адаптер JavaScript.

Оба протокола, *Audience* и *Audience Resolve* , добавляют аудитории только в токен доступа по умолчанию. Токен идентификатора обычно содержит только одну аудиторию, идентификатор клиента, для которого был выдан токен, что является требованием спецификации OpenID Connect. Однако токен доступа не обязательно содержит идентификатор клиента, для которого был выдан токен, если только его не добавили картографы аудитории.

Создание SAML-клиента

Tuxedo SSO поддерживает SAML 2.0 для зарегистрированных приложений. Поддерживаются привязки POST и Redirect. Вы можете выбрать требование проверки подписи клиента. Вы также можете заставить сервер подписывать и/или шифровать ответы.

Процедура

1. Нажмите «Клиенты» в меню.
2. Нажмите «Создать клиента» , чтобы перейти на страницу «Создать клиента» .
3. Установите тип клиента на SAML .

   Создать клиента
4. Введите идентификатор клиента . Часто это URL-адрес, который является ожидаемым значением эмитента в запросах SAML, отправляемых приложением.
5. Нажмите Сохранить . Это действие создаст клиент и перенесет вас на вкладку Настройки .

В следующих разделах описывается каждый параметр на этой вкладке.

Вкладка «Настройки»

Вкладка «Настройки» содержит множество параметров для настройки этого клиента.

Настройки клиента

Общие настройки

Идентификатор клиента

> Буквенно-цифровая строка идентификатора, используемая в запросах OIDC и в базе данных Tuxedo SSO для идентификации клиента. Это значение должно соответствовать значению эмитента, отправленному с AuthNRequests. Tuxedo SSO извлекает эмитента из запроса Authn SAML и сопоставляет его с клиентом по этому значению.

Имя

> Имя клиента на экране пользовательского интерфейса Tuxedo SSO. Чтобы локализовать имя, задайте заменяющее строковое значение. Например, строковое значение, такое как ${myapp}. Для получения дополнительной информации см. Руководство разработчика сервера .

Описание

Описание клиента. Этот параметр также может быть локализован.

Всегда отображать в консоли

Всегда указывать этого клиента в консоли учетной записи, даже если у этого пользователя нет активного сеанса.

Настройки доступа

Корневой URL-адрес

Когда Tuxedo SSO использует настроенный относительный URL-адрес, это значение добавляется к URL-адресу.

Домашняя страница URL

Если Tuxedo SSO необходимо связаться с клиентом, используется этот URL.

Допустимые URI перенаправления

Введите шаблон URL и щелкните знак +, чтобы добавить. Щелкните знак -, чтобы удалить. Щелкните Сохранить , чтобы сохранить эти изменения. Значения подстановочных знаков разрешены только в конце URL. Например, http://host.com/*$$. Это поле используется, когда точные конечные точки SAML не зарегистрированы, и Tuxedo SSO извлекает URL-адрес потребителя утверждений из запроса.

Имя URL-адреса единого входа, инициированного IDP

Имя фрагмента URL для ссылки на клиента, когда вы хотите сделать IDP Initiated SSO. Если оставить это пустым, IDP Initiated SSO будет отключен. URL, на который вы будете ссылаться из своего браузера, будет: server-root /realms/{realm}/protocol/saml/clients/{client-url-name}

Состояние реле SSO, инициированное IDP

Состояние ретранслятора, которое вы хотите отправить с запросом SAML, когда вы хотите выполнить единый вход, инициированный IDP.

Основной URL-адрес обработки SAML

Этот URL используется для всех запросов SAML, а ответ направляется SP. Он используется как URL-адрес Assertion Consumer Service и URL-адрес Single Logout Service.

Если запросы на вход содержат URL-адрес Assertion Consumer Service, то эти запросы на вход будут иметь приоритет. Этот URL-адрес должен быть проверен зарегистрированным шаблоном Valid Redirect URI.

Возможности SAML

Формат идентификатора имени

Формат идентификатора имени для субъекта. Этот формат используется, если в запросе не указана политика идентификатора имени или если атрибут Force Name ID Format установлен в положение ON.

Формат идентификатора имени силы

Если запрос имеет политику идентификатора имени, проигнорируйте ее и используйте значение, настроенное в консоли администратора в разделе Формат идентификатора имени .

Принудительное связывание POST

По умолчанию Tuxedo SSO отвечает, используя начальную привязку SAML исходного запроса. При включении Force POST Binding Tuxedo SSO отвечает, используя привязку SAML POST, даже если исходный запрос использовал привязку перенаправления.

Силовое связывание артефактов

Если эта функция включена, ответные сообщения возвращаются клиенту через систему привязки SAML ARTIFACT.

Включить AuthnStatement

Ответы на вход в систему SAML могут указывать используемый метод аутентификации, например пароль, а также временные метки входа и

истечения сеанса. Включить AuthnStatement включено по умолчанию, поэтому элемент AuthnStatement будет включен в ответы на вход. Установка этого значения в OFF запрещает клиентам определять максимальную продолжительность сеанса, что может создавать клиентские сеансы, которые не истекают.

Включить условие одноразового использования

Если эта опция включена, в ответы на вход в систему включается условие OneTimeUse.

Оптимизация поиска ключа подписи REDIRECT

При установке на ON сообщения протокола SAML включают собственное расширение Tuxedo SSO. Это расширение содержит подсказку с идентификатором ключа подписи. SP использует расширение для проверки подписи вместо попытки проверки подписи с помощью ключей.

Эта опция применяется к привязкам REDIRECT, где подпись передается в параметрах запроса, и эта информация не находится в информации подписи. Это противоречит сообщениям привязки POST, где идентификатор ключа всегда включен в подпись документа.

Эта опция используется, когда сервер Tuxedo SSO и адаптер предоставляют IDP и SP. Эта опция актуальна только тогда, когда Sign Documents установлен на ON.

Подпись и шифрование

Подписывать документы

Если установлено значение ВКЛ, Tuxedo SSO подписывает документ, используя закрытый ключ области.

Подпишите утверждения

Утверждение подписано и встроено в ответ SAML XML Auth.

Алгоритм подписи

Алгоритм, используемый при подписании документов SAML. Обратите внимание, что SHA1основанные алгоритмы устарели и могут быть удалены в будущем выпуске. Мы рекомендуем использовать какой-либо более безопасный алгоритм вместо *_SHA1. Кроме того, с *_SHA1алгоритмами проверка подписей не работает, если клиент SAML работает на Java 17 или выше.

Имя ключа подписи SAML

Подписанные документы SAML, отправленные с помощью привязки POST, содержат идентификацию ключа подписи в элементе KeyName . Это действие можно контролировать с помощью параметра SAML Signature Key Name . Этот параметр управляет содержимым Keyname .

- KEY_ID KeyName содержит идентификатор ключа. Эта опция является опцией по умолчанию .
- CERT_SUBJECT KeyName содержит субъект из сертификата , соответствующего ключу области. Эта опция ожидается службами федерации Microsoft Active Directory.
- НЕТ Подсказка KeyName полностью исключена из сообщения SAML.

Метод канонизации

Метод канонизации для XML-подписей.

Настройки входа

Тема входа

Тема для использования на страницах входа в систему, одноразовых паролей, предоставления регистрации и восстановления забытого пароля.

Требуется согласие

Если эта функция включена, пользователи должны дать согласие на клиентский доступ.

Для клиентов на стороне клиента, которые выполняют вход в браузер. Поскольку невозможно гарантировать, что секреты будут храниться в безопасности с помощью клиентов на стороне клиента, важно ограничить доступ, настроив правильные URI перенаправления.

Отображение клиента на экране

Этот переключатель применяется, если параметр «Требуется согласие» отключен .

- Выключенный

  Экран согласия будет содержать только согласия, соответствующие настроенным клиентским областям действия.

- На

  На экране согласия также будет один пункт, касающийся самого клиента.

Текст экрана согласия клиента

Применяется, если включены параметры Consent required и Display client on screen . Содержит текст, который будет отображаться на экране согласия о разрешениях для этого клиента.

Настройки выхода

Выход из переднего канала

Если Front Channel Logout включен, приложению требуется перенаправление браузера для выполнения выхода. Например, приложению может потребоваться сброс cookie, что можно сделать только с помощью перенаправления. Если Front Channel Logout отключен, Tuxedo SSO вызывает фоновый запрос SAML для выхода из приложения.

Вкладка «Ключи»

Шифровать утверждения

Шифрует утверждения в документах SAML с помощью закрытого ключа областей. Алгоритм AES использует размер ключа 128 бит.

Требуется подпись клиента

Если включен параметр Client Signature Required , ожидается, что документы, поступающие от клиента, будут подписаны. Tuxedo SSO проверит эту подпись, используя открытый ключ клиента или сертификат, настроенный на Keysвкладке.

Разрешить поток ECP

Если значение true, этому приложению разрешено использовать профиль SAML ECP для аутентификации.

Вкладка «Дополнительно»

Эта вкладка содержит много полей для определенных ситуаций. Некоторые поля рассматриваются в других темах. Для получения подробной информации о других полях щелкните значок вопросительного знака.

Детальная конфигурация конечной точки SAML
URL-адрес логотипа

URL-адрес, ссылающийся на логотип клиентского приложения.

URL-адрес политики

URL-адрес, который Клиент проверяющей стороны предоставляет Конечному пользователю для прочтения информации о том, как будут использоваться данные профиля.

URL-адрес условий обслуживания

URL-адрес, который Клиент Проверяющей стороны предоставляет Конечному пользователю для ознакомления с условиями обслуживания Проверяющей стороны.

Утверждение Потребитель Сервис POST Привязка URL

URL-адрес привязки POST для службы потребителей утверждений.

URL-адрес привязки перенаправления службы потребителя утверждения

Перенаправление URL-адреса привязки для службы потребителей утверждений.

URL-адрес привязки POST-запроса службы выхода из системы

URL-адрес привязки POST для службы выхода из системы.

URL-адрес привязки перенаправления службы выхода из системы

Перенаправление привязки URL для службы выхода из системы.

URL-адрес привязки артефакта службы выхода из системы

URL-адрес привязки артефакта для службы выхода из системы. При установке вместе с Force Artifact Bindingопцией
привязка артефакта принудительно применяется как для входа, так и для выхода из системы. Привязка артефакта не используется для выхода из системы, если это свойство не установлено.

URL-адрес привязки SOAP-службы выхода из системы

URL-адрес привязки перенаправления для службы выхода из системы. Применимо только при использовании выхода из системы по обратному каналу .

URL-адрес привязки артефакта

URL-адрес для отправки сообщений HTTP-артефакта.

Служба разрешения артефактов

URL-адрес конечной точки SOAP клиента, куда следует
отправлять ArtifactResolvecообщения.

Вход, инициированный IDP

IDP Initiated Login — это функция, которая позволяет вам настроить конечную точку на сервере Tuxedo SSO, которая будет регистрировать вас в определенном приложении/клиенте. На вкладке Settings для вашего клиента вам необходимо указать URL-имя IDP Initiated SSO . Это простая строка без пробелов. После этого вы можете ссылаться на своего клиента по следующему URL-адресу:root/realms/{realm}/protocol/saml/clients/{url-name}

Реализация входа, инициированная IDP,
предпочитает привязку POST вместо REDIRECT (для получения дополнительной информации проверьте привязки saml). Поэтому окончательная привязка и URL SP выбираются следующим образом:

1. Если определен конкретный URL-адрес привязки POST-запроса Assertion Consumer Service (внутри раздела Fine Grain SAML Endpoint Configuration в настройках клиента), привязка POST используется через этот URL-адрес.
2. Если указан общий URL-адрес обработки основного SAML , то привязка POST снова используется по всему этому общему URL-адресу.
3. В крайнем случае, если настроен URL-адрес перенаправления привязки Assertion Consumer Service (внутри Fine Grain SAML Endpoint Configuration), с этим URL-адресом используется привязка REDIRECT .

Если ваш клиент требует особого состояния реле, вы также можете настроить его на вкладке Настройки в поле Состояние реле, инициированное IDP SSO . В качестве альтернативы браузеры могут указать состояние реле в параметре запроса RelayStateroot/realms/{realm}/protocol/saml/clients/{url-name}?RelayState=thestate , т. е . .

При использовании брокера идентификации можно настроить IDP Initiated Login для клиента из внешнего IDP. Фактический клиент настраивается для IDP Initiated Login в брокерском IDP, как описано выше. Внешний IDP должен настроить клиента для приложения IDP Initiated Login, который будет указывать на специальный URL, указывающий на брокера и представляющий конечную точку

IDP Initiated Login для выбранного клиента в брокерском IDP. Это означает, что в настройках клиента на внешнем IDP:

- Имя URL-адреса единого входа, инициированного IDP, задается как имя, которое будет опубликовано в качестве начальной точки входа, инициированного IDP.
- URL-адрес привязки POST-запроса службы потребителя утверждений в разделе «Конфигурация конечной точки SAML с подробным описанием» должен быть установлен на следующий URL-адрес: broker-root/realms/{broker-realm}/broker/{idp-name}/endpoint/clients/{client-id}, где:
    - broker-root — это базовый URL-адрес брокера
    - broker-realm — имя области у брокера, где объявлен внешний IDP
    - idp-name — имя внешнего IDP у брокера
    - client-id — это значение атрибута IDP Initiated SSO URL Name клиента SAML, определенного на брокере. Это тот клиент, который будет доступен для IDP Initiated Login из внешнего IDP.

Обратите внимание, что вы можете импортировать базовые настройки клиента из брокерского IDP в настройки клиента внешнего IDP — просто используйте дескриптор SP, доступный в настройках поставщика удостоверений в брокерском IDP, и добавьте его к URL-адресу конечной точки.clients/client-id

Использование дескриптора сущности для создания клиента

Вместо регистрации клиента SAML 2.0 вручную вы можете импортировать клиент с помощью стандартного XML-файла дескриптора сущности SAML.

На странице «Клиент» имеется опция «Импорт клиента» .

Добавить клиента

Процедура

1. Нажмите «Обзор» .

2. Загрузите файл, содержащий информацию о дескрипторе сущности XML.
3. Проверьте информацию, чтобы убедиться, что все настроено правильно.

Некоторые клиентские адаптеры SAML, такие как mod-auth-mellon , требуют XML Entity Descriptor для IDP. Вы можете найти этот дескриптор, перейдя по этому URL:

root/realms/{realm}/protocol/saml/descriptor

где realm — область вашего клиента.

Клиентские ссылки

Для связи одного клиента с другим Tuxedo SSO предоставляет конечную точку перенаправления: /realms/realm_name/clients/{client-id}/redirect.

Если клиент обращается к этой конечной точке с помощью HTTP GETзапроса, Tuxedo SSO возвращает настроенный базовый URL для предоставленного Клиента и Области в форме HTTP 307(временного перенаправления) в заголовке ответа Location. В результате этого клиенту нужно знать только имя Области и Идентификатор Клиента, чтобы ссылаться на них. Это косвенное обращение позволяет избежать жесткого кодирования клиентских базовых URL.

В качестве примера рассмотрим область действия masterи идентификатор клиента account:

http://host:port/realms/master/clients/account/redirect

Этот URL временно перенаправляет на: http://host:port/realms/master/account

Сопоставление токенов OIDC и утверждений SAML

Приложения, получающие токены идентификации, токены доступа или утверждения SAML, могут требовать разные роли и метаданные пользователя.

Вы можете использовать Tuxedo SSO для:

- Жестко запрограммируйте роли, утверждения и пользовательские атрибуты.
- Извлечь метаданные пользователя в токен или утверждение.
- Переименовать роли.

Эти действия выполняются на вкладке «Картографы» в консоли администратора.

Вкладка «Картографы»

Новые клиенты не имеют встроенных картографов, но они могут наследовать некоторые карты из клиентских областей. Подробнее см. в разделе клиентские области .

Протокольные преобразователи сопоставляют элементы (например, адрес электронной почты) с определенным утверждением в идентификаторе и токене доступа. Функция преобразователя должна быть самоочевидной из его названия. Вы добавляете предварительно настроенные преобразователи, нажимая Добавить встроенный .

Каждый картограф имеет набор общих настроек. Дополнительные настройки доступны в зависимости от типа картографа. Нажмите «Изменить» рядом с картографом, чтобы открыть экран конфигурации и настроить эти настройки.

Конфигурация картографа

Подробную информацию о каждой опции можно просмотреть, наведя курсор на ее подсказку.

Вы можете использовать большинство картографов OIDC для управления размещением заявки. Вы можете включить или исключить заявку из идентификатора и токенов доступа , настроив переключатели Добавить в токен ID и Добавить в токен доступа .

Вы можете добавлять типы картографов следующим образом:

Процедура

1. Перейдите на вкладку «Картографы» .
2. Нажмите Настроить новый картограф .

   Добавить картографа

3. Выберите тип картографа из списка.

Приоритетный порядок

Реализации Mapper имеют приоритетный порядок . Приоритетный порядок не является свойством конфигурации Mapper. Это свойство конкретной реализации Mapper.

Mappers сортируются по порядку в списке mappers. Изменения в токене или утверждении применяются в этом порядке, начиная с самого низкого. Таким образом, реализации, зависящие от других реализаций, обрабатываются в необходимом порядке.

Например, чтобы вычислить роли, которые будут включены в токен:

1. Распределите аудитории на основе этих ролей.
2. Обработать скрипт JavaScript, который использует роли и аудитории, уже имеющиеся в токене.

Картографы заметок сеансов пользователей OIDC

Подробности сеанса пользователя определяются с помощью картографов и автоматически включаются при использовании или включении функции на клиенте. Нажмите Добавить встроенный , чтобы включить подробности сеанса.

Сеансы персонифицированного пользователя предоставляют следующую информацию:

- IMPERSONATOR_ID : идентификатор пользователя, выдающего себя за другого человека.
- IMPERSONATOR_USERNAME : Имя пользователя, выдающего себя за другого пользователя.

Сеансы учетной записи службы предоставляют следующую информацию:

- clientId : идентификатор клиента учетной записи службы.
- client_id : идентификатор клиента учетной записи службы.

- clientAddress : IP-адрес удаленного хоста аутентифицированного устройства учетной записи службы.
- clientHost : имя удаленного хоста аутентифицированного устройства учетной записи службы.

Скрипт-картограф

Используйте Script Mapper для сопоставления заявок с токенами путем запуска пользовательского кода JavaScript. Для получения более подробной информации о развертывании скриптов на сервере см. JavaScript Providers .

При развертывании скриптов вы должны иметь возможность выбирать развернутые скрипты из списка доступных картографов.

Парный картограф идентификаторов субъектов

Утверждение субъекта sub сопоставляется по умолчанию сопоставителем протокола Subject (sub) в базовой области клиента по умолчанию .

Чтобы использовать парный идентификатор субъекта с помощью сопоставителя протоколов, такого как Pairwise Subject Identifier , вы можете удалить сопоставитель протоколов Subject (sub) из базовой клиентской области. Однако это не является строго необходимым, так как сопоставитель протоколов Subject (sub) выполняется до сопоставителя идентификаторов Pairwise субъектов , и, следовательно, парное значение переопределит значение, добавленное сопоставителем Subject. Это связано с приоритетом сопоставителя Subject. Таким образом, единственным преимуществом удаления встроенного сопоставителя Subject (sub) может быть небольшая экономия производительности за счет отказа от использования сопоставителя протоколов, который может не иметь никакого эффекта.

Использование облегченного токена доступа

Токен доступа в Tuxedo SSO содержит конфиденциальную информацию, включая персональную идентифицируемую информацию (PII). Поэтому, если сервер ресурсов не хочет раскрывать этот тип информации третьим лицам, таким как клиенты, Tuxedo SSO поддерживает легкие токены доступа, которые удаляют PII

из токенов доступа. Кроме того, когда сервер ресурсов получает PII, удаленный из токена доступа, он может получить PII, отправив токен доступа в конечную точку интроспекции токенов Tuxedo SSO.

Информация, которую невозможно удалить из облегченного токена доступа

> Протокольные картографы могут контролировать, какая информация помещается в токен доступа, а легкий токен доступа использует протокольные картографы. Поэтому следующая информация не может быть удалена из легкого доступа.
> exp, iat, jti, iss, typ, azp, sid, scope,cnf

Использование облегченного токена доступа в Tuxedo SSO

> Применяя к клиенту политикуuse-lightweight-access-token исполнителя клиента , клиент может получить облегченный токен доступа вместо токена доступа. Облегченный токен доступа содержит заявку, контролируемую сопоставителем протоколов, где его настройка (по умолчанию ВЫКЛ) включена. Кроме того, включив настройку сопоставителя протоколов, клиент может получить заявку, отправив токен доступа в конечную точку интроспекции токенов Tuxedo SSO.Add to lightweight access tokenAdd to token introspection

Конечная точка интроспекции

> В некоторых случаях может быть полезно активировать конечную точку интроспекции токена с заголовком HTTP Accept: application/jwtвместо Accept: application/json, что может быть полезно, особенно для легких токенов доступа. Подробности о конечной точке интроспекции токена см. в разделе о защите приложений .

Генерация конфигурации клиентского адаптера

Tuxedo SSO может генерировать файлы конфигурации, которые можно использовать для установки клиентского адаптера в среде развертывания вашего приложения. Для OIDC и SAML поддерживается ряд типов адаптеров.

1. Нажмите меню «Действие» и выберите опцию «Загрузить конфигурацию адаптера».
2. Выберите параметр формата, для которого вы хотите сгенерировать конфигурацию.

Поддерживаются все клиентские адаптеры Tuxedo SSO для OIDC и SAML. Поддерживается адаптер Apache HTTPD mod-auth-mellon для SAML, а также стандартные файлы дескрипторов сущностей SAML.

Области применения клиента

Используйте Tuxedo SSO для определения общей клиентской конфигурации в сущности, называемой клиентской областью . Клиентская область настраивает сопоставления протоколов и сопоставления областей ролей для нескольких клиентов.

Клиентские области также поддерживают параметр области OAuth 2. Клиентские приложения используют этот параметр для запроса утверждений или ролей в токене доступа в зависимости от требований приложения.

Чтобы создать клиентскую область, выполните следующие действия:

1. В меню выберите «Области действия клиента» .

   Список областей клиента

2. Нажмите «Создать» .
3. Назовите область деятельности вашего клиента.
4. Нажмите «Сохранить ».

Область действия клиента имеет вкладки, похожие на вкладки обычных клиентов. Вы можете определить сопоставители протоколов и сопоставления областей действия ролей . Эти сопоставления могут наследоваться другими клиентами и настроены на наследование из этой области действия клиента.

Протокол

При создании клиентской области выберите Протокол . Клиенты, связанные в одной области, должны иметь один и тот же протокол.

Каждая область имеет набор предопределенных встроенных клиентских областей в меню.

- Протокол SAML: role_list . Эта область содержит один сопоставитель протоколов для списка ролей в утверждении SAML.
- Протокол OpenID Connect: доступно несколько клиентских областей:
    - роли

      Эта область не определена в спецификации OpenID Connect и не добавляется автоматически в заявку области в токене доступа. Эта область имеет сопоставители, которые используются для добавления ролей пользователя в токен доступа и добавления аудиторий для клиентов, имеющих хотя бы одну роль клиента. Эти сопоставители более подробно описаны в разделе Аудитория .

    - веб-источники

      Эта область также не определена в спецификации OpenID Connect и не добавлена в область, заявляющую токен доступа. Эта область используется для добавления разрешенных веб-источников в заявку на токен доступа allowed-origins .

    - микропрофиль-jwt

      Эта область обрабатывает утверждения, определенные в спецификации MicroProfile/JWT Auth . Эта область определяет средство сопоставления свойств пользователя для утверждения upn и средство сопоставления ролей области для утверждения groups . Эти средства сопоставления можно изменять, чтобы можно было использовать различные свойства для создания определенных утверждений MicroProfile/JWT.

    - офлайн_доступ

Эта область используется в случаях, когда клиентам необходимо получить офлайн-токены. Более подробная информация об офлайн-токенах доступна в разделе «Офлайн-доступ» и в спецификации OpenID Connect .

- профиль
- электронная почта
- адрес
- телефон

Профиль клиентских областей , email , адрес и телефон определены в спецификации OpenID Connect . Для этих областей не определены сопоставления областей ролей, но для них определены сопоставители протоколов. Эти сопоставители соответствуют утверждениям, определенным в спецификации OpenID Connect.

Например, если открыть область действия телефонного клиента и открыть вкладку «Сопоставители» , вы увидите сопоставители протоколов, которые соответствуют утверждениям, определенным в спецификации для области действия телефона .

Картографы клиентской области

Когда область телефонного клиента связана с клиентом, клиент автоматически наследует все сопоставители протоколов, определенные в области телефонного клиента. Маркеры доступа, выданные для этого клиента, содержат информацию о номере телефона пользователя, предполагая, что у пользователя есть определенный номер телефона.

Встроенные клиентские области содержат сопоставители протоколов, как определено в спецификации. Вы можете свободно редактировать клиентские области и создавать, обновлять или удалять любые сопоставители протоколов или сопоставления областей ролей.

Настройки, связанные с согласием

Области клиента содержат опции, связанные с экраном согласия. Эти опции полезны, если связанный клиент, если Consent Required включен на клиенте.

Отображение на экране согласия

> Если включено Отображение на экране согласия и область добавлена к клиенту, требующему согласия, текст, указанный в Тексте экрана согласия, будет отображаться на экране согласия. Этот текст отображается, когда пользователь проходит аутентификацию и до того, как пользователь перенаправляется из Tuxedo SSO на клиент. Если отключено Отображение на экране согласия , эта область клиента не будет отображаться на экране согласия.

Текст экрана согласия

> Текст, отображаемый на экране согласия, когда эта клиентская область добавляется к клиенту, когда требуется согласие, по умолчанию соответствует имени клиентской области. Значение этого текста можно настроить, указав переменную подстановки со строками ${var-name} . Настраиваемое значение настраивается в файлах свойств в вашей теме. Дополнительные сведения о настройке см. в руководстве разработчика сервера .

Свяжите область действия клиента с клиентом

Связывание между клиентской областью и клиентом настраивается на вкладке Client Scopes клиента. Доступны два способа связывания клиентской области и клиента.

Области действия клиента по умолчанию

> Этот параметр применим к клиентам OpenID Connect и SAML. Области действия клиента по умолчанию применяются при выдаче токенов OpenID Connect или утверждений SAML для клиента. Клиент унаследует сопоставления протоколов и сопоставления областей ролей, определенные в области клиента. Для протокола OpenID Connect сопоставления и сопоставления областей ролей применяются всегда, независимо от значения, используемого для параметра области действия в запросе авторизации OpenID Connect.

Дополнительные клиентские области

> Эта настройка применима только для клиентов OpenID Connect. Необязательные клиентские области применяются при выдаче токенов для этого клиента, но только если это запрошено параметром области в запросе авторизации OpenID Connect.

Пример

Для этого примера предположим, что у клиента есть профиль и адрес электронной почты, связанные как области клиента по умолчанию, а
также телефон и адрес, связанные как необязательные области клиента. Клиент использует значение параметра области при отправке запроса в конечную точку авторизации OpenID Connect.

scope=openid phone

Параметр scope содержит строку со значениями scope, разделенными пробелами. Значение openid — это мета-значение, используемое для всех запросов OpenID Connect. Токен будет содержать сопоставления и сопоставления областей ролей из профиля клиентских областей по умолчанию и email , а также phone — необязательная область клиента, запрошенная параметром scope.

Оценка клиентских возможностей

Вкладка **Mappers** содержит сопоставителей протоколов, а вкладка **Scope** содержит сопоставления областей ролей, объявленные для этого клиента. Они не содержат сопоставлений и сопоставлений областей, унаследованных от клиентских областей. Можно увидеть эффективные сопоставления протоколов (то есть сопоставления протоколов, определенные на самом клиенте, а также унаследованные от связанных клиентских областей) и эффективные сопоставления областей ролей, используемые при генерации токена для клиента.

Процедура

1. Откройте вкладку «Области действия клиента» для нужного клиента.
2. Откройте вложенную вкладку « Оценить» .
3. Выберите дополнительные клиентские области, которые вы хотите применить.

Это также покажет вам значение параметра scope . Этот параметр необходимо отправить из приложения в конечную точку авторизации Tuxedo SSO OpenID Connect.

Оценка клиентских возможностей

Чтобы отправить пользовательское значение для параметра **области действия** из вашего приложения, см. **адаптер JavaScript Tuxedo SSO** в разделе «Защита приложений» для адаптеров JavaScript.

Все примеры генерируются для конкретного пользователя и выдаются для конкретного клиента с указанным значением параметра scope . Примеры включают все используемые утверждения и сопоставления ролей.

Разрешения клиентской области

При выдаче токенов пользователю область действия клиента применяется только в том случае, если пользователю разрешено его использовать.

Если в клиентской области не определены сопоставления областей ролей, каждому пользователю разрешено использовать эту клиентскую область. Однако если в клиентской области определены сопоставления областей ролей, пользователь должен быть членом хотя бы одной из ролей. Должно быть пересечение между ролями пользователя и ролями клиентской области. Составные роли учитываются при оценке этого пересечения.

Если пользователю не разрешено использовать область действия клиента, то при генерации токенов не будут использоваться ни сопоставители протоколов, ни сопоставления областей действия ролей. Область действия клиента не будет отображаться в значении области действия токена.

Области клиента Realm по умолчанию

Используйте **области действия клиентов по умолчанию Realm** для определения наборов областей действия клиентов, которые автоматически связываются с вновь созданными клиентами.

Процедура

1. Откройте вкладку «Области действия клиента» для нужного клиента.

Здесь выберите клиентские области, которые вы хотите добавить в качестве клиентских областей по умолчанию для вновь созданных клиентов и дополнительных клиентских областей .

Области действия клиента по умолчанию

При создании клиента вы можете отсоединить области действия клиента по умолчанию, если это необходимо. Это похоже на удаление ролей по умолчанию .

Объяснение областей применения

Область действия клиента

> Клиентские области — это сущности в Tuxedo SSO, которые настраиваются на уровне области и могут быть связаны с клиентами. Клиентские области ссылаются по их имени, когда запрос отправляется на конечную точку авторизации Tuxedo SSO с соответствующим значением параметра области . Более подробную информацию см. в разделе связывания клиентских областей .

Отображение области действия ролей

> Это доступно на вкладке Область действия клиента или области действия клиента. Используйте сопоставление области действия ролей , чтобы ограничить роли, которые могут использоваться в токенах доступа. Более подробную информацию см. в разделе Сопоставления области действия ролей .

Области действия авторизации

> Область авторизации охватывает действия, которые могут быть выполнены в приложении. Более подробную информацию см. в Руководстве по службам авторизации .

Политика в отношении клиентов

Чтобы упростить защиту клиентских приложений, полезно реализовать следующие пункты унифицированным образом.

- Установка политик относительно того, какую конфигурацию может иметь клиент
- Проверка клиентских конфигураций
- Соответствие требуемым стандартам безопасности и профилям, таким как Financial-grade API (FAPI) и OAuth 2.1

Для комплексной реализации этих положений введена концепция клиентской политики .

Варианты использования

Клиентская политика реализует следующие пункты, указанные ниже.

Установка политик относительно того, какую конфигурацию может иметь клиент

> Параметры конфигурации на клиенте могут быть принудительно применены политиками клиента во время создания/обновления клиента, а также во время запросов OpenID Connect к серверу Tuxedo SSO, которые связаны с конкретным клиентом. Tuxedo SSO поддерживает аналогичные вещи также через политики регистрации клиентов, описанные в службе регистрации клиентов из Руководств по защите приложений . Однако политики регистрации клиентов могут охватывать только динамическую регистрацию клиентов OIDC. Политики клиента охватывают не только то, что могут делать политики регистрации клиентов, но и другие способы регистрации и настройки клиентов. Текущие планы заключаются в замене регистрации клиентов политиками клиента.

Проверка клиентских конфигураций

> Tuxedo SSO поддерживает проверку того, следует ли клиент настройкам, таким как Proof Key for Code Exchange, Request Object Signing Algorithm, Holder-of-Key Token и т. д., для некоторых конечных точек, таких как Authorization Endpoint, Token Endpoint и т. д. Они могут быть указаны каждым элементом настройки (в консоли администратора, переключателе, раскрывающемся меню и т. д.). Чтобы сделать клиентское приложение безопасным, администратору необходимо установить множество настроек

соответствующим образом, что затрудняет для администратора задачу по защите клиентского приложения. Клиентские политики могут выполнять эти проверки клиентских конфигураций, упомянутые выше, и их также можно использовать для автоматической настройки некоторых переключателей клиентской конфигурации для соответствия расширенным требованиям безопасности. В будущем отдельные параметры конфигурации клиента могут быть заменены клиентскими политиками, которые напрямую выполняют требуемые проверки.

Соответствие требуемым стандартам безопасности и профилям, таким как FAPI и OAuth 2.1

Глобальные профили клиентов — это профили клиентов, предварительно настроенные в Tuxedo SSO по умолчанию. Они предварительно настроены для соответствия стандартным профилям безопасности, таким как FAPI и OAuth 2.1 в разделе защиты приложений , что позволяет администратору легко защитить свое клиентское приложение для соответствия определенному профилю безопасности. На данный момент Tuxedo SSO имеет глобальные профили для поддержки спецификаций FAPI и OAuth 2.1. Администратору нужно будет просто настроить клиентские политики, чтобы указать, какие клиенты должны соответствовать FAPI и OAuth 2.1. Администратор может настроить клиентские профили и клиентские политики, так что клиенты Tuxedo SSO можно будет легко сделать совместимыми с различными другими профилями безопасности, такими как SPA, Native App, Open Banking и т. д.

Протокол

Концепция клиентской политики не зависит от какого-либо конкретного протокола. В настоящее время Tuxedo SSO поддерживает, в частности, клиентские профили для протокола OpenID Connect (OIDC) , но также доступен клиентский профиль для протокола SAML .

Архитектура

Политика клиента состоит из четырех основных блоков: Условие, Исполнитель, Профиль и Политика.

Состояние
Условие определяет, к какому клиенту принимается политика и когда она принимается. Некоторые условия проверяются во время создания/обновления клиента, когда некоторые другие условия проверяются во время клиентских запросов (запрос авторизации OIDC, запрос конечной точки токена и т. д.). Условие проверяет, удовлетворен ли один из указанных критериев. Например, некоторые условия проверяют, является ли тип доступа клиента конфиденциальным.

Условие не может быть использовано само по себе. Оно может быть использовано в политике , которая описана далее.

Условие может быть настраиваемым так же, как и другие настраиваемые поставщики. То, что можно настроить, зависит от природы каждого условия.

Предусмотрены следующие условия:

Способ создания/обновления клиента

- Динамическая регистрация клиента (анонимная или аутентифицированная с помощью начального токена доступа или токена доступа к регистрации)
- API REST администратора (консоль администратора и т. д.)

Так, например, при создании клиента можно настроить условие для оценки как истинное, когда этот клиент создан с помощью динамической регистрации клиентов OIDC без начального токена доступа (анонимная динамическая регистрация клиентов). Так, например, это условие можно использовать для обеспечения того, чтобы все клиенты, зарегистрированные с помощью динамической регистрации клиентов OIDC, были совместимы с FAPI или OAuth 2.1.

Автор клиента (проверяется по присутствию в определенной роли или группе)

> При динамической регистрации клиента OpenID Connect автором клиента является конечный пользователь, который был аутентифицирован для получения токена доступа для генерации нового клиента, а не учетная запись службы существующего клиента, которая фактически получает доступ к

конечной точке регистрации с помощью токена доступа. При регистрации через Admin REST API автором клиента является конечный пользователь, например администратор Tuxedo SSO.

Тип клиентского доступа (конфиденциальный, публичный, только на предъявителя)

Например, когда клиент отправляет запрос на авторизацию, политика принимается, если этот клиент является конфиденциальным. Конфиденциальный клиент включил аутентификацию клиента, когда публичный клиент отключил аутентификацию клиента. Bearer-only — устаревший тип клиента.

Область действия клиента

Оценивается как true, если у клиента есть определенная клиентская область (либо как область по умолчанию, либо как необязательная область, используемая в текущем запросе). Это может быть использовано, например, для обеспечения того, чтобы запросы авторизации OIDC с областью fapi-example-scopeбыли совместимы с FAPI.

Роль клиента

Применяется для клиентов с клиентской ролью указанного имени. Обычно вы можете создать клиентскую роль указанного имени для запрошенных клиентов и использовать ее как «маркерную роль», чтобы убедиться, что указанная клиентская политика будет применена для запрошенных клиентов.

Часто существует вариант использования для требования применения определенной клиентской политики для указанных клиентов, таких как `my-client-1`и `my-client-2`. Лучший способ достичь этого результата — использовать условие **Client Role** в вашей политике, а затем создать клиентскую роль с указанным именем для запрошенных клиентов. Эту клиентскую роль можно использовать как «роль маркера», используемую исключительно для маркировки этой конкретной клиентской политики для определенных клиентов.

Доменное имя клиента, хост или IP-адрес

Применяется для определенных доменных имен клиента. Или для случаев, когда администратор регистрирует/обновляет клиента с определенного хоста или IP-адреса.

Атрибут клиента

> Применяется к клиентам с атрибутом клиента указанного имени и значения. Если указать несколько атрибутов клиента, они будут оцениваться с использованием условий AND. Если вы хотите оценивать с использованием условий OR, задайте это условие несколько раз.

Любой клиент

> Это условие всегда оценивается как истинное. Его можно использовать, например, для обеспечения того, чтобы все клиенты в определенной области были совместимы с FAPI.

Исполнитель

Исполнитель указывает, какое действие выполняется на клиенте, к которому применяется политика. Исполнитель выполняет одно или несколько указанных действий. Например, некоторые исполнители проверяют, redirect_uricовпадает ли значение параметра в запросе авторизации с одним из предварительно зарегистрированных URI перенаправления на конечной точке авторизации, и отклоняют этот запрос, если нет.

Исполнитель не может быть использован сам по себе. Он может быть использован в профиле , который описан далее.

Исполнитель может быть настроен так же, как и другие настраиваемые поставщики. То, что можно настроить, зависит от природы каждого исполнителя.

Исполнитель действует на различные события. Реализация исполнителя может игнорировать определенные типы событий (например, исполнитель для проверки requestобъекта OIDC действует только на запрос авторизации OIDC). Событиями являются:

- Создание клиента (включая создание посредством динамической регистрации клиента)
- Обновление клиента
- Отправка запроса на авторизацию

- Отправка запроса токена
- Отправка запроса на обновление токена
- Отправка запроса на отзыв токена
- Отправка запроса на интроспекцию токена
- Отправка запроса на информацию о пользователе
- Отправка запроса на выход из системы с токеном обновления (обратите внимание, что выход из системы с токеном обновления — это фирменная функция Tuxedo SSO, не поддерживаемая ни одной спецификацией. Вместо этого рекомендуется полагаться на официальный выход из системы OIDC).

На каждом событии исполнитель может работать в несколько фаз. Например, при создании/обновлении клиента исполнитель может изменить конфигурацию клиента, автоматически настроив определенные параметры клиента. После этого исполнитель проверяет эту конфигурацию на этапе проверки.

Одной из нескольких целей этого исполнителя является реализация требований безопасности клиентских профилей соответствия, таких как FAPI и OAuth 2.1. Для этого необходимы следующие исполнители:

- Для клиента используется метод Enforce secure Client Authentication
- Используются токены Enforce Holder-of-key
- Используется ключ подтверждения для обмена кодами (PKCE)
- Используется алгоритм защищенной подписи для аутентификации клиента Signed JWT (private-key-jwt)
- Принудительно используйте URI перенаправления HTTPS и убедитесь, что настроенный URI перенаправления не содержит подстановочных знаков
- Обеспечить requestобъект OIDC, удовлетворяющий высокому уровню безопасности
- Принудительно применять тип ответа гибридного потока OIDC, включая токен ID, используемый в качестве отдельной подписи , как описано в

спецификации FAPI 1. Это означает, что токен ID, возвращаемый из ответа авторизации, не будет содержать данные профиля пользователя.

- Обеспечить более безопасную stateобработку nonceпараметров для предотвращения CSRF-атак
- Применять более безопасный алгоритм подписи при регистрации клиента
- Параметр Enforce binding_messageиспользуется для запросов CIBA
- Обеспечить ротацию секретной информации клиента
- Обеспечить наличие токена доступа к регистрации клиента
- Принудительно проверять, является ли клиент тем, которому было выдано намерение, в случае использования, когда намерение выдается до начала потока кода авторизации, чтобы получить токен доступа, как в UK OpenBanking.
- Обеспечить запрет неявного и гибридного потока
- Принудительно проверять, содержит ли запрос PAR необходимые параметры, включенные в запрос авторизации
- Используются токены принудительной привязки DPoPdpop (доступно, если функция включена)
- Обеспечить использование облегченного токена доступа
- Обеспечить пропуск ротации токенов обновления и отсутствие возврата токенов обновления из ответа на токен обновления
- Обеспечить наличие допустимого URI перенаправления, требуемого спецификацией OAuth 2.1
- Принудительное связывание SAML Redirect не может быть использовано или запросы и утверждения SAML подписаны

Профиль

Профиль состоит из нескольких исполнителей, которые могут реализовать профиль безопасности, такой как FAPI и OAuth 2.1. Профиль может быть настроен с помощью Admin REST API (Admin Console) вместе с его исполнителями.

Существует три глобальных профиля , и они настроены в Tuxedo SSO по умолчанию с предварительно настроенными исполнителями, соответствующими спецификациям FAPI 1 Baseline, FAPI 1 Advanced, FAPI CIBA, FAPI 2 и OAuth 2.1. Более подробная информация содержится в разделе FAPI и OAuth 2.1 в разделе приложений безопасности .

Политика

Политика состоит из нескольких условий и профилей. Политика может быть принята к клиентам, удовлетворяющим всем условиям этой политики. Политика ссылается на несколько профилей, и все исполнители этих профилей выполняют свою задачу в отношении клиента, к которому принята эта политика.

Конфигурация

Политики, профили, условия, исполнители могут быть настроены с помощью Admin REST API, что также означает Admin Console. Для этого есть вкладка Realm → Realm Settings → Client Policies , что означает, что администратор может иметь клиентские политики для каждой области.

Глобальные профили клиентов автоматически доступны в каждой области. Однако по умолчанию не настроены клиентские политики. Это означает, что администратору всегда требуется создавать любую клиентскую политику, если он хочет, например, чтобы клиенты его области соответствовали FAPI. Глобальные профили не могут быть обновлены, но администратор может легко использовать их в качестве шаблона и создать свой собственный профиль, если он хочет внести некоторые незначительные изменения в конфигурации глобального профиля. В консоли администратора доступен редактор JSON, который упрощает создание нового профиля на основе некоторого глобального профиля.

Обратная совместимость

Политики клиента могут заменить политики регистрации клиента, описанные в службе регистрации клиента из Руководств по защите приложений . Однако политики регистрации клиента также по-прежнему сосуществуют. Это означает, что, например, во время запроса динамической регистрации клиента для

создания/обновления клиента применяются как политики клиента, так и политики регистрации клиента.

В настоящее время планируется удалить функцию политик регистрации клиентов, а существующие политики регистрации клиентов будут автоматически перенесены в новые политики клиентов.

Пример ротации секрета клиента

См. пример конфигурации для ротации клиентских секретов .

Использование хранилища для получения секретов

В настоящее время Tuxedo SSO предоставляет две готовые реализации Vault SPI: хранилище на основе обычного текстового файла и хранилище на основе Java KeyStore.

Чтобы получить секрет из хранилища, а не вводить его напрямую, введите следующую специально созданную строку в соответствующее поле:

${vault.key}

где key— имя секрета, распознанного хранилищем.

Чтобы предотвратить утечку секретов между областями, Tuxedo SSO объединяет имя области с keyполученным из выражения хранилища. Этот метод означает, что keyне сопоставляется напрямую с записью в хранилище, а создает окончательное имя записи в соответствии с алгоритмом, используемым для объединения keyс именем области. В случае хранилища на основе файлов такая комбинация отражает определенное имя файла, для хранилища на основе Java KeyStore это определенное имя псевдонима.

Вы можете получить секрет из хранилища в следующих полях:

SMTP-пароль

В настройках SMTP- сервера

Учетные данные привязки LDAP

В настройках LDAP федерации пользователей на основе LDAP.

Секрет поставщика удостоверений OIDC

В Client Secret внутри поставщика удостоверений OpenID Connect Config

Ключевые решатели

Все встроенные поставщики поддерживают настройку ключевых распознавателей. Ключевой распознаватель реализует алгоритм или стратегию для объединения имени области с ключом, полученным из выражения ${vault.key}, в конечное имя записи, используемое для извлечения секрета из хранилища. Tuxedo SSO использует keyResolverscвойство для настройки распознавателей, используемых поставщиком. Значение представляет собой разделенный запятыми список имен распознавателей. Ниже приведен пример конфигурации для поставщика files-plaintext:

```
kc.[sh|bat] start --spi-vault-file-key-
resolvers=REALM_UNDERSCORE_KEY,KEY_ONLY
```

Решатели запускаются в том же порядке, в котором вы объявляете их в конфигурации. Для каждого решателя Tuxedo SSO использует последнее имя записи, созданное решателем, которое объединяет область с ключом хранилища для поиска секрета хранилища. Если Tuxedo SSO находит секрет, он возвращает секрет. Если нет, Tuxedo SSO использует следующий решатель. Этот поиск продолжается до тех пор, пока Tuxedo SSO не найдет непустой секрет или не закончатся решатели. Если Tuxedo SSO не находит секрета, Tuxedo SSO возвращает пустой секрет.

В предыдущем примере Tuxedo SSO REALM_UNDERSCORE_KEYсначала использует решатель. Если Tuxedo SSO находит запись в хранилище, которая использует этот решатель, Tuxedo SSO возвращает эту запись. Если нет, Tuxedo SSO снова ищет с помощью KEY_ONLYрешателя. Если Tuxedo SSO находит запись с помощью KEY_ONLYрешателя, Tuxedo SSO возвращает эту запись. Если Tuxedo SSO использует все решатели, Tuxedo SSO возвращает пустой секрет.

Ниже приведен список доступных на данный момент резолверов:

Имя	Описание
ТОЛЬКО_КЛЮЧ	Tuxedo SSO игнорирует имя области и использует ключ из выражения хранилища.
REALM_UNDERSCORE_KEY	Tuxedo SSO объединяет область и ключ, используя символ подчеркивания. Tuxedo SSO экранирует вхождения подчеркивания в области или ключе с помощью другого символа подчеркивания. Например, если область вызывается `master_realm`и ключ — `smtp_key`, объединенный ключ — `master__realm_smtp__key`.
REALM_FILESEPARATOR_KEY	Tuxedo SSO объединяет область и ключ, используя символ-разделитель файлов платформы.
ПРЕДОСТАВЛЕНО ЗАВОДОМ	Tuxedo SSO объединяет область и ключ, используя фабрику поставщика хранилища `VaultKeyResolver`, что позволяет создать пользовательский распознаватель ключей путем расширения существующей фабрики и реализации `getFactoryResolver`метода.

Если вы не настроили резолвер для встроенных поставщиков, Tuxedo SSO выбирает REALM_UNDERSCORE_KEY.

Настройка аудита для отслеживания событий

Tuxedo SSO включает в себя набор возможностей аудита. Вы можете записывать каждый вход в систему и действие администратора и просматривать эти действия в консоли администратора. Tuxedo SSO также включает в себя SPI-прослушиватель, который прослушивает события и может запускать действия. Примеры встроенных прослушивателей включают файлы журналов и отправку электронных писем при возникновении события.

Аудит пользовательских событий

Вы можете записывать и просматривать каждое событие, которое влияет на пользователей. Tuxedo SSO запускает события входа для таких действий, как успешный вход пользователя, ввод пользователем неверного пароля или обновление учетной записи пользователя. По умолчанию Tuxedo SSO не сохраняет

и не отображает события в консоли администратора. В консоли администратора и в файле журнала сервера регистрируются только события ошибок.

Процедура

Используйте эту процедуру, чтобы начать аудит пользовательских событий.

1. Нажмите «Настройки области» в меню.
2. Перейдите на вкладку «События» .
3. Перейдите на вкладку Настройки пользовательских событий .
4. Установите переключатель Сохранение событий в положение ВКЛ .

   Настройки пользовательских событий
5. Укажите срок хранения событий в поле «Срок действия» .
6. Нажмите «Добавить сохраненные типы», чтобы увидеть другие события, которые вы можете сохранить.

   Добавить типы
7. Нажмите Добавить .

Нажмите Очистить события пользователя, если вы хотите удалить все сохраненные события.

Процедура

Теперь вы можете просматривать события.

1. Нажмите вкладку «События» в меню.

   Пользовательские события
2. Чтобы отфильтровать события, нажмите Поиск пользовательских событий .

   Поиск события пользователя

Типы событий

События входа:

Событие	Описание
Авторизоваться	Пользователь входит в систему.
Зарегистрироваться	Пользователь регистрируется.
Выйти	Пользователь выходит из системы.
Код для токена	Приложение или клиент обменивает код на токен.
Обновить токен	Приложение или клиент обновляет токен.

Защита от грубой силы:

Событие	Описание
Пользователь отключен постоянной блокировкой	Защита от атак методом подбора пароля навсегда отключила учетную запись пользователя из-за слишком большого количества неудачных попыток входа в систему.
Пользователь отключен из-за временной блокировки	Защита от атак методом подбора пароля временно отключила учетную запись пользователя из-за слишком большого количества неудачных попыток входа в систему.

Посредничество в идентификации:

Событие	Описание
Переопределение ссылки федеративной идентификации	Существующая ссылка на федеративную идентификацию была переопределена
Ошибка переопределения ссылки федеративной идентификации	Произошла ошибка при попытке переопределить существующую ссылку на федеративную идентификацию

OAuth:

Событие	Описание
Предоставление разрешения на расширение OAuth2	Грант OAuth2 был выполнен
Ошибка предоставления расширения OAuth2	Произошла ошибка во время выполнения гранта OAuth2

События аккаунта:

Событие	Описание
Социальная ссылка	Учетная запись пользователя связана с поставщиком социальных сетей.
Удалить социальную ссылку	Связь между аккаунтом социальной сети и аккаунтом пользователя прерывается.
Обновить электронную почту	Изменился адрес электронной почты для учетной записи.
Обновить профиль	Изменяется профиль учетной записи.
Отправить сброс пароля	Tuxedo SSO отправляет электронное письмо для сброса пароля.
Обновить пароль (устарело)	Пароль к учетной записи меняется.
Обновить учетные данные	Настройки пароля или одноразового пароля (OTP/TOTP) для учетной записи изменяются.
Обновление TOTP (устарело)	Изменены настройки одноразового пароля (TOTP) для учетной записи.
Удалить TOTP (устарело)	Tuxedo SSO удаляет TOTP из аккаунта.
Удалить учетные данные	Tuxedo SSO удаляет учетные данные из учетной записи.
Отправить Подтвердить адрес электронной почты	Tuxedo SSO отправляет электронное письмо с подтверждением.
Подтвердить адрес электронной почты	Tuxedo SSO проверяет адрес электронной почты для учетной записи.

Каждому событию соответствует событие ошибки.

Прослушиватель событий

Слушатели событий прослушивают события и выполняют действия на основе этого события. Tuxedo SSO включает два встроенных прослушивателя: прослушиватель событий регистрации и прослушиватель событий электронной почты.

Прослушиватель событий регистрации

Если включен прослушиватель событий регистрации, этот прослушиватель записывает данные в файл журнала при возникновении события ошибки.

Пример сообщения журнала от прослушивателя событий регистрации:

11:36:09,965 ПРЕДУПРЕЖДЕНИЕ [org.Tuxedo SSO.events] (задача по умолчанию-51) тип=LOGIN_ERROR, realmId=master,
clientId=myapp,
userId=19aeb848-96fc-44f6-b0a3-59a17570d374,
ipAddress=127.0.0.1,
ошибка=неверные_учетные_данные_пользователя,
метод_аутентификации=openid-connect, тип_аутентификации=код,
redirect_uri=http://localhost:8180/myapp,
code_id=b669da14-cdbb-41d0-b055-0810a0334607, имя пользователя=admin

Вы можете использовать прослушиватель событий регистрации для защиты от атак хакерских ботов:

1. Проанализируйте файл журнала на предмет LOGIN_ERRORсобытия.
2. Извлеките IP-адрес события неудачного входа в систему.
3. Отправьте IP-адрес в программный инструмент предотвращения вторжений.

Logging Event Listener регистрирует события в org.Tuxedo SSO.eventsкатегории журнала. Tuxedo SSO по умолчанию не включает события журнала отладки в журналы сервера.

Чтобы включить события журнала отладки в журналы сервера:

1. Изменить уровень журнала для org.Tuxedo SSO.eventsкатегории
2. Измените уровень ведения журнала, используемый прослушивателем событий регистрации.

Чтобы изменить уровень журнала, используемый прослушивателем событий регистрации, добавьте следующее:

bin/kc.[sh|bat] start --spi-events-listener-jboss-logging-success-level=info --spi-events-listener-jboss-logging-error-level=error

Допустимые значения для уровней журнала: debug, info, warn, errorи fatal.

Прослушиватель событий электронной почты
Прослушиватель событий электронной почты отправляет сообщение на адрес электронной почты пользователя при возникновении события и поддерживает следующие события:

- Ошибка входа.
- Обновить пароль.
- Обновите одноразовый пароль с ограниченным сроком действия (TOTP).
- Удалить одноразовый пароль (OTP).
- Обновите учетные данные.
- Удалить учетные данные.

Для отправки электронного письма необходимо соблюдение следующих условий:

- У пользователя есть адрес электронной почты.
- Адрес электронной почты пользователя отмечен как проверенный.

Предпосылки

- Настройки электронной почты Realm настроены.

Процедура

Чтобы включить прослушиватель электронной почты:

1. Нажмите «Настройки области» в меню.
2. Перейдите на вкладку «События» .
3. Щелкните поле «Прослушиватели событий» .
4. Выбирать email.

   Прослушиватели событий

Вы можете исключить события, используя --spi-events-listener-email-exclude-eventsаргумент. Например:

kc.[sh|bat] --spi-events-listener-email-exclude-events=UPDATE_CREDENTIAL,REMOVE_CREDENTIAL

Аудит административных событий

Вы можете записывать все действия, выполняемые администратором в консоли администратора. Консоль администратора выполняет административные действия, вызывая интерфейс REST Tuxedo SSO, а Tuxedo SSO проверяет эти вызовы REST. Вы можете просматривать результирующие события в консоли администратора.

Процедура

Используйте эту процедуру, чтобы начать аудит действий администратора.

1. Нажмите «Настройки области» в меню.
2. Перейдите на вкладку «События» .
3. Откройте вкладку «Настройки событий администратора» .
4. Установите переключатель Сохранение событий в положение ВКЛ .

   Tuxedo SSO отображает переключатель представления «Включить» .
5. Переключите параметр Включить представление в положение ВКЛ .

   Коммутатор Include Representationвключает документы JSON, отправляемые через REST API администратора, что позволяет просматривать действия администраторов.

   Настройки событий администратора
6. Нажмите «Сохранить ».
7. Чтобы очистить базу данных сохраненных действий, нажмите Очистить административные события .

Процедура

Теперь вы можете просматривать события администратора.

1. Нажмите «События» в меню.

2. Откройте вкладку «События администратора» .

   Административные события

Когда Include Representationпереключатель находится в положении ON, это может привести к хранению большого объема информации в базе данных. Вы можете задать максимальную длину представления, используя --spi-events-store-jpa-max-field-lengthаргумент. Эта настройка полезна, если вы хотите придерживаться базового ограничения хранения. Например:

kc.[sh|bat] --spi-events-store-jpa-max-field-length=2500

Снижение угроз безопасности

Уязвимости безопасности существуют в любом сервере аутентификации. Для получения дополнительной информации см. Модель угроз OAuth 2.0 от Internet Engineering Task Force (IETF) и Лучшая текущая практика безопасности OAuth 2.0 .

Хозяин

Tuxedo SSO использует публичное имя хоста несколькими способами, например, в полях эмитента токена и URL-адресах в письмах для сброса пароля.

По умолчанию имя хоста выводится из заголовков запроса. Не существует проверки, гарантирующей допустимость имени хоста. Если вы не используете балансировщик нагрузки или прокси с Tuxedo SSO для предотвращения недопустимых заголовков хоста, настройте приемлемые имена хостов.

Интерфейс поставщика услуг (SPI) имени хоста предоставляет способ настройки имени хоста для запросов. Вы можете использовать этого встроенного поставщика для установки фиксированного URL для запросов frontend, одновременно разрешая запросы backend на основе URI запроса. Если встроенный поставщик не имеет требуемых возможностей, вы можете разработать настраиваемого поставщика.

Конечные точки администратора и консоль администратора

Tuxedo SSO выставляет административный REST API и веб-консоль на том же порту, что и неадминистративное использование. Не выставляйте административные конечные точки наружу, если внешний доступ не нужен.

Атаки методом грубой силы

Атака методом подбора пароля пытается угадать пароль пользователя, пытаясь войти в систему несколько раз. Tuxedo SSO имеет возможности обнаружения методом подбора пароля и может временно отключить учетную запись пользователя, если количество неудачных попыток входа превысит указанный порог.

> Tuxedo SSO отключает обнаружение brute force по умолчанию. Включите эту функцию для защиты от brute force атак.

Процедура

Чтобы включить эту защиту:

1. Нажмите «Настройки области» в меню.
2. Перейдите на вкладку «Защита» .
3. Нажмите вкладку «Обнаружение методом подбора» .

Обнаружение методом грубой силы

Tuxedo SSO может применять действия постоянной и временной блокировки при обнаружении атаки. Постоянная блокировка отключает учетную запись пользователя до тех пор, пока администратор не включит ее повторно. Временная блокировка отключает учетную запись пользователя на определенный период времени. Период времени, в течение которого учетная запись отключена, увеличивается по мере продолжения атаки, а последующие сбои достигают кратности Max Login Failures.

> Когда пользователь временно заблокирован и пытается войти в систему, Tuxedo SSO

отображает `Invalid username or password`сообщение об ошибке по умолчанию. Это сообщение является тем же сообщением об ошибке, что и сообщение, отображаемое при недопустимом имени пользователя или недопустимом пароле, чтобы гарантировать, что злоумышленник не знает, что учетная запись отключена.

Общие параметры

Имя	Описание	По умолчанию
Макс. количество неудачных попыток входа	Максимальное количество неудачных попыток входа в систему.	30 неудач.
Быстрая проверка входа в систему в миллисекундах	Минимальное время между попытками входа в систему.	1000 миллисекунд.
Минимальное время ожидания быстрого входа	Минимальное время, в течение которого пользователь будет отключен, если попытки входа в систему выполняются быстрее, чем *проверка быстрого входа в систему в миллисекундах* .	1 минута.

Параметры временной блокировки

Имя	Описание	По умолчанию
Ожидание приращения	Время, добавляемое к времени временного отключения пользователя, когда количество попыток входа пользователя превышает *максимальное количество неудачных попыток входа* .	1 минута.
Макс Подожди	Максимальное время временного отключения пользователя.	15 минут.
Время сброса отказа	Время, когда счетчик неудач сбрасывается. Таймер запускается с последнего неудачного входа. Убедитесь, что это число всегда больше `Max wait`; в противном случае эффективное время ожидания никогда не достигнет установленного вами значения `Max wait`.	12 часов.

Алгоритм временной блокировки

1. При успешном входе в систему

a. Перезагрузитьcount

2. При неудачном входе в систему

a. Если время между этим сбоем и последним сбоем больше, чем время сброса сбоя

i. Перезагрузитьcount

b. Приращениеcount

c. Рассчитать waitс помощью Wait Increment * (count/ Max Login Failures). Деление представляет собой целочисленное деление, округленное до целого числа

d. Если waitравно 0 и время между этим сбоем и последним сбоем меньше, чем Миллисекунды проверки быстрого входа , установите waitМинимальное ожидание быстрого входа .

i. Временно отключить пользователя на минимальное из waitи максимальное время ожидания секунд

ii. Увеличить счетчик временных блокировок

countне увеличивается, если временно отключенная учетная запись совершает ошибку входа в систему.

Например, если вы установили Max Login Failuresзначение 5и a Wait Incrementсекунд 30, эффективное время, в течение которого учетная запись будет отключена после нескольких неудачных попыток аутентификации, составит:

Number of Failures	Wait Increment	Max Login Failures	Effective Wait Time
1	30	5	0
2	30	5	0
3	30	5	0
4	30	5	0

5	**30**	5	**30**
6	30	5	30
7	30	5	30
8	30	5	30
9	30	5	30
10	**30**	5	**60**

Обратите внимание, что Effective Wait Timeпри 5-й неудачной попытке аккаунт будет отключен на 30секунд. Только после достижения следующего кратного Max Login Failures, в данном случае 10, время увеличится с 30до 60. Время, в течение которого аккаунт будет отключен, увеличивается только при достижении кратного Max Login Failures.

Параметры постоянной блокировки

Имя	Описание	По умолчанию
Максимальное количество временных блокировок	Максимально допустимое количество временных блокировок до наступления постоянной блокировки.	0

Постоянный поток блокировки

1. Следуйте временному потоку блокировки
2. Если счетчик временных блокировок превышает максимальное количество временных блокировок
    a. Навсегда отключить пользователя

Когда Tuxedo SSO отключает пользователя, пользователь не может войти в систему, пока администратор не включит пользователя. Включение учетной записи сбрасывает count.

Недостатком обнаружения методом подбора Tuxedo SSO является то, что сервер становится уязвимым для атак типа «отказ в обслуживании». При реализации

атаки типа «отказ в обслуживании» злоумышленник может попытаться войти в систему, угадывая пароли для любых известных ему учетных записей, и в конечном итоге заставит Tuxedo SSO отключить учетные записи.

Рассмотрите возможность использования программного обеспечения для предотвращения вторжений (IPS). Tuxedo SSO регистрирует все неудачные попытки входа и неудачные попытки IP-адреса клиента. Вы можете указать IPS на файл журнала сервера Tuxedo SSO, и IPS может изменять брандмауэры для блокировки подключений с этих IP-адресов.

Политика паролей

Убедитесь, что у вас есть сложная политика паролей, чтобы заставить пользователей выбирать сложные пароли. Для получения дополнительной информации см. главу Политики паролей . Предотвратите угадывание паролей, настроив сервер Tuxedo SSO на использование одноразовых паролей.

Атрибуты пользователя только для чтения

Типичные пользователи, хранящиеся в Tuxedo SSO, имеют различные атрибуты, связанные с их профилями пользователей. К таким атрибутам относятся email, firstName или lastName. Однако пользователи также могут иметь атрибуты, которые не являются типичными данными профиля, а скорее метаданными. Атрибуты метаданных обычно должны быть доступны только для чтения для пользователей, и типичные пользователи никогда не должны иметь возможности обновить эти атрибуты из пользовательского интерфейса Tuxedo SSO или API REST учетной записи. Некоторые атрибуты должны быть доступны только для чтения даже для администраторов при создании или обновлении пользователя с помощью API REST администратора.

Атрибуты метаданных обычно представляют собой атрибуты из следующих групп:

- Различные ссылки или метаданные, связанные с поставщиками хранения данных пользователя. Например, в случае интеграции LDAP LDAP_IDатрибут содержит идентификатор пользователя на сервере LDAP.

- Метаданные, предоставленные User Storage.
  Например, createdTimestampпредоставленные из LDAP, должны быть всегда доступны только для чтения пользователем или администратором.
- Метаданные, связанные с различными аутентификаторами.
  Например, KERBEROS_PRINCIPALатрибут может содержать имя принципала kerberos конкретного пользователя. Аналогично атрибут usercertificateможет содержать метаданные, связанные со связыванием пользователя с данными из сертификата X.509, который обычно используется, когда включена аутентификация по сертификату X.509.
- Метаданные, связанные с идентификатором пользователей приложениями/клиентами.
  Например, saml.persistent.name.id.for.my_appмогут содержать SAML NameID, который будет использоваться клиентским приложением my_appв качестве идентификатора пользователя.
- Метаданные, связанные с политиками авторизации, которые используются для управления доступом на основе атрибутов (ABAC). Значения этих атрибутов могут использоваться для решений об авторизации. Поэтому важно, чтобы эти атрибуты не могли быть обновлены пользователями.

В долгосрочной перспективе Tuxedo SSO будет иметь надлежащий профиль пользователя SPI, который позволит выполнять тонкую настройку каждого атрибута пользователя. В настоящее время эта возможность еще не полностью доступна. Поэтому Tuxedo SSO имеет внутренний список атрибутов пользователя, которые доступны только для чтения для пользователей и только для чтения для администраторов, настроенных на уровне сервера.

Это список атрибутов, доступных только для чтения, которые используются внутренними средствами поставщиков и функций Tuxedo SSO по умолчанию и, следовательно, всегда доступны только для чтения:

- Для
  пользователей: KERBEROS_PRINCIPAL, LDAP_ID, LDAP_ENTRY_DN, CR

EATED_TIMESTAMP, createTimestamp, modifyTimestamp, userCertificate, saml.persistent.name.id.for.*, ENABLED,EMAIL_VERIFIED

- Для администраторов: KERBEROS_PRINCIPAL, LDAP_ID, LDAP_ENTRY_DN, CREATED_TIMESTAMP, createTimestamp,modifyTimestamp

Системные администраторы имеют возможность добавлять дополнительные атрибуты в этот список. Конфигурация в настоящее время доступна на уровне сервера.

Эту конфигурацию можно добавить с помощью опций spi-user-profile-declarative-user-profile-read-only-attributesи `spi-user-profile-declarative-user-profile-admin-read-only-attributes. Например:

kc.[sh|bat] start --spi-user-profile-declarative-user-profile-read-only-attributes=foo,bar*

В этом примере пользователи и администраторы не смогут обновить атрибут foo. Пользователи не смогут редактировать атрибуты, начинающиеся с bar. Например, barили barrier. Конфигурация нечувствительна к регистру, поэтому атрибуты типа FOOили BarRiertакже будут отклонены в этом примере. Подстановочный знак *поддерживается только в конце имени атрибута, поэтому администратор может фактически отклонить все атрибуты, начинающиеся с указанного символа. *В середине атрибута считается обычным символом.

Проверка атрибутов пользователя

С помощью функций управления атрибутами пользователей администраторы могут ограничивать данные, которые пользователи вводят для атрибутов, например, при регистрации пользователя или в консоли учетной записи.

Администраторы не должны разрешать пользователям неуправляемые атрибуты, чтобы помешать злоумышленникам добавлять неограниченное количество атрибутов. Атрибуты должны иметь проверку, которая ограничивает объем данных, вводимых злоумышленниками.

При использовании регулярных выражений для проверки атрибутов пользователя избегайте регулярных выражений, которые используют чрезмерное количество памяти или ЦП. Подробности см. в OWASP Regular expression Denial of Service .

Кликджекинг

Clickjacking — это метод обмана пользователей, заставляющий их нажимать на элемент пользовательского интерфейса, отличный от того, который воспринимают пользователи. Вредоносный сайт загружает целевой сайт в прозрачном iFrame, наложенном поверх набора фиктивных кнопок, размещенных прямо под важными кнопками на целевом сайте. Когда пользователь нажимает видимую кнопку, он нажимает кнопку на скрытой странице. Злоумышленник может украсть учетные данные аутентификации пользователя и получить доступ к его ресурсам, используя этот метод.

По умолчанию каждый ответ Tuxedo SSO устанавливает некоторые конкретные заголовки HTTP, которые могут предотвратить это. В частности, он устанавливает X-Frame-Options и Content-Security-Policy . Вам следует взглянуть на определение обоих этих заголовков, поскольку существует множество тонкого доступа к браузеру, который вы можете контролировать.

Процедура

В консоли администратора вы можете указать значения заголовков X-Frame-Options и Content-Security-Policy.

1. Нажмите на пункт меню «Настройки области» .
2. Перейдите на вкладку «Защита» .

   Безопасность Защита

По умолчанию Tuxedo SSO устанавливает политику единого источника только для iframe.

Требование SSL/HTTPS

OAuth 2.0/OpenID Connect использует токены доступа для обеспечения безопасности. Злоумышленники могут сканировать вашу сеть на наличие токенов доступа и использовать их для выполнения вредоносных операций, на которые у токена есть разрешение. Эта атака известна как атака типа «человек посередине». Используйте SSL/HTTPS для связи между сервером аутентификации Tuxedo SSO и клиентами, которых Tuxedo SSO защищает для предотвращения атак типа «человек посередине».

Tuxedo SSO имеет три режима для SSL/HTTPS . SSL сложен в настройке, поэтому Tuxedo SSO позволяет осуществлять связь без HTTPS через частные IP-адреса, такие как localhost, 192.168.xx и другие частные IP-адреса. В производстве убедитесь, что вы включили SSL, и SSL является обязательным для всех операций.

На стороне адаптера/клиента вы можете отключить менеджер доверия SSL. Менеджер доверия гарантирует, что идентификация клиента, с которым взаимодействует Tuxedo SSO, является действительной, и проверяет доменное имя DNS на соответствие сертификату сервера. В производстве убедитесь, что каждый из ваших клиентских адаптеров использует хранилище доверия для предотвращения атак DNS man-in-the-middle.

CSRF-атаки

Атака с подделкой межсайтовых запросов (CSRF) использует HTTP-запросы от пользователей, которые уже прошли аутентификацию на веб-сайтах. Любой сайт, использующий аутентификацию на основе cookie, уязвим для атак CSRF. Вы можете смягчить эти атаки, сопоставив cookie состояния с опубликованной формой или параметром запроса.

Спецификация входа OAuth 2.0 требует, чтобы cookie-файл состояния совпадал с переданным параметром состояния. Tuxedo SSO полностью реализует эту часть спецификации, поэтому все входы защищены.

Tuxedo SSO Admin Console — это приложение JavaScript/HTML5, которое делает вызовы REST к бэкэнду Tuxedo SSO admin REST API. Все эти вызовы требуют аутентификации токена носителя и состоят из вызовов JavaScript Ajax, поэтому

CSRF невозможен. Вы можете настроить admin REST API для проверки источников CORS.

Консоль учетной записи в Tuxedo SSO может быть уязвима для CSRF. Чтобы предотвратить атаки CSRF, Tuxedo SSO устанавливает файл cookie состояния и встраивает значение этого файла cookie в скрытые поля формы или параметры запроса в ссылках действий. Tuxedo SSO проверяет параметр запроса/формы по файлу cookie состояния, чтобы убедиться, что вызов был сделан тем же пользователем.

Неопределенные URI перенаправления

Сделайте зарегистрированные URI перенаправления максимально конкретными. Регистрация неопределенных URI перенаправления для потоков кода авторизации может позволить вредоносным клиентам выдавать себя за другого клиента с более широким доступом. Выдача себя за другого клиента может произойти, например, если два клиента находятся в одном домене.

Вы можете использовать secure redirect uris forcer executor для вашей области. Результат гарантирует, что администраторы клиентов смогут регистрировать только клиентов с определенными redirect-uris, соответствующими различным требованиям, таким как требование, чтобы URL не имел подстановочных знаков в пути контекста или мог быть ограничен указанными разрешенными доменами. См. Client Policies для получения подробной информации о том, как настроить клиентские политики с определенным executor.

Соответствие FAPI

Чтобы убедиться, что сервер Tuxedo SSO проверит ваш клиент на предмет большей безопасности и соответствия FAPI, вы можете настроить клиентские политики для поддержки FAPI. Подробности FAPI описаны в разделе «Безопасность приложений» . Помимо прочего, это обеспечивает некоторые рекомендации по безопасности, описанные выше, такие как SSL,

требуемый для клиентов, используемый URI безопасного перенаправления и другие подобные рекомендации.

Соответствие OAuth 2.1

Чтобы убедиться, что сервер Tuxedo SSO проверит ваш клиент на предмет большей безопасности и совместимости с OAuth 2.1, вы можете настроить клиентские политики для поддержки OAuth 2.1. Подробности OAuth 2.1 описаны в разделе «Защита приложений» .

Скомпрометированный доступ и токены обновления

Tuxedo SSO включает несколько действий, чтобы предотвратить кражу токенов доступа и обновления злоумышленниками. Ключевым действием является обеспечение связи SSL/HTTPS между Tuxedo SSO и его клиентами и приложениями. Tuxedo SSO не включает SSL по умолчанию.

Другим действием по смягчению ущерба от утечки токенов доступа является сокращение срока действия токена. Вы можете указать срок действия токенов на странице тайм-аутов . Короткий срок действия токенов доступа заставляет клиентов и приложения обновлять свои токены доступа через короткое время. Если администратор обнаруживает утечку, он может выйти из всех сеансов пользователей, чтобы сделать эти токены обновления недействительными или настроить политику отзыва.

Убедитесь, что токены обновления всегда остаются конфиденциальными для клиента и никогда не передаются.

Вы можете смягчить ущерб от утечки токенов доступа и обновления токенов, выпустив эти токены как токены держателя ключа. См. OAuth 2.0 Mutual TLS Client Certificate Bound Access Token для получения дополнительной информации.

Если токен доступа или токен обновления скомпрометирован, войдите в консоль администратора и примените политику отзыва not-before ко всем приложениям. Применив политику not-before, вы гарантируете, что все токены, выпущенные до

этого времени, станут недействительными. Применив новую политику not-before, вы гарантируете, что приложения должны загрузить новые открытые ключи из Tuxedo SSO и смягчить ущерб от скомпрометированного ключа подписи области. Дополнительную информацию см. в главе о ключах .

Вы можете отключить определенные приложения, клиентов или пользователей, если они скомпрометированы.

Скомпрометированный код авторизации

Для OIDC Auth Code Flow Tuxedo SSO генерирует криптографически сильное случайное значение для своих кодов авторизации. Код авторизации используется только один раз для получения токена доступа.

На странице тайм-аутов в консоли администратора можно указать длительность действия кода авторизации. Убедитесь, что длительность составляет менее 10 секунд, что достаточно для того, чтобы клиент запросил токен из кода.

Вы также можете защититься от утечки кодов авторизации, применив к клиентам Proof Key for Code Exchange (PKCE) .

Открытые редиректоры

Открытый редиректор — это конечная точка, использующая параметр для автоматического перенаправления агента пользователя в местоположение, указанное значением параметра без проверки. Злоумышленник может использовать конечную точку авторизации конечного пользователя и параметр URI перенаправления, чтобы использовать сервер авторизации в качестве открытого редиректора, используя доверие пользователя к серверу авторизации для запуска фишинговой атаки.

Tuxedo SSO требует, чтобы все зарегистрированные приложения и клиенты регистрировали по крайней мере один шаблон URI перенаправления. Когда клиент запрашивает, чтобы Tuxedo SSO выполнил перенаправление, Tuxedo SSO проверяет URI перенаправления по списку допустимых зарегистрированных

шаблонов URI. Клиенты и приложения должны регистрировать как можно более конкретный шаблон URI, чтобы смягчить атаки открытого перенаправления.

Если приложению требуется не http(s) пользовательская схема, она должна быть явной частью шаблона проверки (например custom:/app/*). По соображениям безопасности общий шаблон вроде *не охватывает не http(s) схемы.

Используя клиентские политики , администратор может гарантировать, что клиенты не смогут регистрировать открытые URL-адреса перенаправления, такие как *.

База данных паролей скомпрометирована

Tuxedo SSO не хранит пароли в виде сырого текста, а как хэшированный текст, используя PBKDF2-HMAC-SHA512алгоритм дайджеста сообщений. Tuxedo SSO выполняет 210,000итерации хэширования, количество итераций, рекомендованное сообществом безопасности. Это количество итераций хэширования может отрицательно сказаться на производительности, поскольку хэширование PBKDF2 использует значительный объем ресурсов ЦП.

Ограничение сферы действия

По умолчанию новые клиентские приложения имеют неограниченное role scope mappings. Каждый токен доступа для этого клиента содержит все разрешения, которые есть у пользователя. Если злоумышленник скомпрометирует клиент и получит токены доступа клиента, каждая система, к которой пользователь может получить доступ, будет скомпрометирована.

Ограничьте роли токена доступа, используя меню Scope для каждого клиента. В качестве альтернативы вы можете установить сопоставления областей ролей на уровне Client Scope и назначить Client Scope вашему клиенту, используя меню Client Scope .

Ограничить аудиторию токенов

В средах с низким уровнем доверия между сервисами ограничьте аудиторию на токене. Для получения дополнительной информации см. Модель угроз OAuth2 и раздел Поддержка аудитории .

Ограничить сеансы аутентификации

Сеансы аутентификации отслеживают состояние аутентификации. Текст ниже применим независимо от исходного потока.

В этом разделе описываются развертывания, в которых для сеансов аутентификации используется поставщик Infinispan.

Сеанс аутентификации хранится внутри как RootAuthenticationSessionEntity. Каждый RootAuthenticationSessionEntityможет иметь несколько подсеансов аутентификации, хранящихся в RootAuthenticationSessionEntityкак коллекция AuthenticationSessionEntityобъектов. Tuxedo SSO хранит сеансы аутентификации в выделенном кэше Infinispan.
Количество AuthenticationSessionEntityper RootAuthenticationSessionEntityвлияет на размер каждой записи кэша. Общий объем памяти кэша сеанса аутентификации определяется количеством сохраненных RootAuthenticationSessionEntityи количеством AuthenticationSessionEntityв каждом RootAuthenticationSessionEntity.

Количество поддерживаемых RootAuthenticationSessionEntityобъектов соответствует количеству незавершенных потоков входа из браузера. Чтобы держать количество RootAuthenticationSessionEntityпод контролем, рекомендуется использовать расширенный контроль брандмауэра для ограничения входящего сетевого трафика.

Более высокое использование памяти может возникнуть для развертываний, где есть много активных RootAuthenticationSessionEntityс большим количеством AuthenticationSessionEntity. Если балансировщик нагрузки не поддерживает или не настроен для прилипания сеанса, нагрузка по сети в кластере может значительно возрасти. Причина этой нагрузки в том, что каждый запрос, который попадает на узел, не владеющий соответствующим сеансом аутентификации, должен извлекать и обновлять запись сеанса аутентификации на

узле-владельце, что подразумевает отдельную сетевую передачу как для извлечения, так и для хранения.

Максимальное
количество AuthenticationSessionEntityper RootAuthenticationSessionEntityможно настроить в authenticationSessionsSPI, установив property authSessionsLimit. Значение по умолчанию установлено на 300 AuthenticationSessionEntityper a RootAuthenticationSessionEntity. При достижении этого предела самая старая подсессия аутентификации будет удалена после запроса новой сессии аутентификации.

В следующем примере показано, как ограничить количество активных элементов AuthenticationSessionEntityна a RootAuthenticationSessionEntityдо 100.

bin/kc.[sh|bat] start --spi-authentication-sessions-infinispan-auth-sessions-limit=100

Эквивалентная команда для нового хранилища карт:

bin/kc.[sh|bat] start --spi-authentication-sessions-map-auth-sessions-limit=100

Атаки с использованием SQL-инъекций

В настоящее время у Tuxedo SSO нет известных уязвимостей SQL-инъекций.

Консоль аккаунта

Пользователи Tuxedo SSO могут управлять своими аккаунтами через Account Console. Они могут настраивать свои профили, добавлять двухфакторную аутентификацию, включать аккаунты поставщиков удостоверений и контролировать активность устройств.

Дополнительные ресурсы

- Account Console можно настроить с точки зрения внешнего вида и языковых предпочтений. Примером может служить добавление дополнительных

атрибутов на страницу Personal info . Для получения дополнительной информации см. Server Developer Guide .

Доступ к консоли учетной записи

Процедура

1. Запишите имя области и IP-адрес сервера Tuxedo SSO, на котором находится ваша учетная запись.
2. В веб-браузере введите URL-адрес в следующем формате: server-root /realms/{realm-name}/account.
3. Введите свое имя пользователя и пароль.

Консоль аккаунта

Настройка способов входа

Вы можете войти в эту консоль, используя базовую аутентификацию (имя пользователя и пароль) или двухфакторную аутентификацию. Для двухфакторной аутентификации используйте одну из следующих процедур.

Двухфакторная аутентификация с OTP

Предпосылки

- OTP — это допустимый механизм аутентификации для вашей области.

Процедура

1. Нажмите в меню « Безопасность учетной записи» .
2. Нажмите «Войти» .
3. Нажмите «Настроить приложение Authenticator» .

   Вход в систему

4. Следуйте инструкциям на экране, чтобы использовать мобильное устройство в качестве генератора одноразовых паролей.
5. Отсканируйте QR-код на снимке экрана и вставьте его в генератор одноразовых паролей на вашем мобильном устройстве.

6. Выйдите из системы и войдите снова.
7. Ответьте на запрос, введя одноразовый пароль, предоставленный на вашем мобильном устройстве.

Двухфакторная аутентификация с WebAuthn

Предпосылки

- WebAuthn — это допустимый механизм двухфакторной аутентификации для вашей области. Пожалуйста, следуйте разделу WebAuthn для получения более подробной информации.

Процедура

1. Нажмите «Безопасность учетной записи» в меню.
2. Нажмите «Войти» .
3. Нажмите «Настроить пароль» .

   Вход в систему

4. Подготовьте свой ключ доступа. То, как вы подготовите этот ключ, зависит от типа используемого вами ключа доступа. Например, для USB-ключей Yubikey вам может потребоваться вставить ключ в USB-порт вашего ноутбука.
5. Нажмите «Зарегистрироваться» , чтобы зарегистрировать свой пароль.
6. Выйдите из системы и войдите снова.
7. Если процесс аутентификации настроен правильно, появится сообщение с просьбой пройти аутентификацию, используя ваш пароль в качестве второго фактора.

Беспарольная аутентификация с помощью WebAuthn

Предпосылки

- WebAuthn — это допустимый механизм аутентификации без пароля для вашей области. Пожалуйста, следуйте разделу Passwordless WebAuthn для получения более подробной информации.

Процедура

1. Нажмите «Безопасность учетной записи» в меню.
2. Нажмите «Войти» .
3. Нажмите «Настроить пароль» в разделе «Без пароля» .

   Вход в систему
4. Подготовьте свой ключ доступа. То, как вы подготовите этот ключ, зависит от типа используемого вами ключа доступа. Например, для USB-ключей Yubikey вам может потребоваться вставить ключ в USB-порт вашего ноутбука.
5. Нажмите «Зарегистрироваться» , чтобы зарегистрировать свой пароль.
6. Выйдите из системы и войдите снова.
7. Если поток аутентификации был настроен правильно, появится сообщение с просьбой пройти аутентификацию с помощью вашего Passkey в качестве второго фактора. Вам больше не нужно указывать свой пароль для входа в систему.

Просмотр активности устройства

Вы можете просматривать устройства, с которых выполнен вход в вашу учетную запись.

Процедура

1. Нажмите в меню « Безопасность учетной записи» .
2. Нажмите Активность устройства .
3. Выйдите из системы, если устройство выглядит подозрительным.

Устройства

Добавление учетной записи поставщика удостоверений

Вы можете связать свой аккаунт с брокером идентификации . Эта опция часто используется для привязки аккаунтов социальных провайдеров.

Процедура

1. Войдите в консоль администратора.
2. В меню выберите Поставщики удостоверений .
3. Выберите поставщика и заполните поля.
4. Вернитесь в консоль аккаунта.
5. Нажмите в меню « Безопасность учетной записи» .
6. Нажмите Связанные учетные записи .

Добавленный вами поставщик удостоверений отображается на этой странице.

Связанные аккаунты

Доступ к другим приложениям

Пункт меню Приложения показывает пользователям, к каким приложениям у вас есть доступ. В этом случае доступна только Консоль учетной записи.

Приложения

Просмотр членства в группах

Вы можете просмотреть группы, с которыми вы связаны, нажав на меню Группы . Если вы установите флажок Прямое членство , вы увидите только группы, с которыми вы связаны напрямую.

Предпосылки

- Для просмотра меню «Группы» вам необходимо иметь роль учетной записи view-groups .

Посмотреть членство в группах

Административный интерфейс командной строки

С помощью Tuxedo SSO вы можете выполнять задачи администрирования из интерфейса командной строки (CLI), используя инструмент командной строки Admin CLI.

Установка административного CLI

Tuxedo SSO упаковывает дистрибутив сервера Admin CLI со скриптами выполнения в binкаталоге.

Скрипт для Linux называется kcadm.sh, а скрипт для Windows называется kcadm.bat. Добавьте каталог сервера Tuxedo SSO в свойPATH чтобы использовать клиент из любого места в вашей файловой системе.

Например:

- Линукс:

  $ экспорт ПУТЬ=$ПУТЬ:$Tuxedo SSO_HOME/bin
  $ kcadm.sh

- Окна:

  c:\> установить PATH=%PATH%;%Tuxedo SSO_HOME%\bin
  c:\> ккадм

Вам необходимо установить `Tuxedo SSO_HOME`переменную среды на путь, куда вы извлекли дистрибутив Tuxedo SSO Server.

Чтобы избежать повторений, в остальной части документа примеры Windows используются только в тех местах, где различия CLI заключаются не только в `kcadm`названии команды.

Использование интерфейса командной строки администратора

Admin CLI делает HTTP-запросы к конечным точкам Admin REST. Для доступа к конечным точкам Admin REST требуется аутентификация.

Подробную информацию об атрибутах JSON для конкретных конечных точек см. в документации по API REST администратора.

1. Начните сеанс аутентификации, войдя в систему. Теперь вы можете выполнять операции создания, чтения, обновления и удаления (CRUD).

Например:

- Линукс:

  $ kcadm.sh учетные данные конфигурации --server http://localhost:8080 --realm demo --user admin --client admin
  $ kcadm.sh создать области -s realm=demorealm -s enabled=true -o
  $ CID=$(kcadm.sh создать клиентов -r demorealm -s clientId=my_client -s 'redirectUris=["http://localhost:8980/myapp/*"]' -i)
  $ kcadm.sh получить клиенты/$CID/установка/провайдеры/Tuxedo SSO-oidc-Tuxedo SSO-json

- Окна:

  c:\> учетные данные конфигурации kcadm --server http://localhost:8080 --realm demo --user admin --client admin
  c:\> kcadm create realms -s realm=demorealm -s enabled=true -o
  c:\> kcadm создать клиентов -r demorealm -s clientId=my_client -s "redirectUris=[\"http://localhost:8980/myapp/*\"]" -i > clientid.txt
  c:\> set /p CID=<clientid.txt
  c:\> kcadm получить клиенты/%CID%/установка/провайдеры/Tuxedo SSO-oidc-Tuxedo SSO-json

2. В производственной среде получите доступ к Tuxedo SSO с помощью , https:чтобы избежать раскрытия токенов. Если доверенный центр сертификации, включенный в хранилище сертификатов Java по умолчанию, не выдал сертификат сервера, подготовьте truststore.jksфайл и дайте команду Admin CLI использовать его.

   Например:

   - Линукс:

     $ kcadm.sh конфигурация truststore --trustpass $ПАРОЛЬ ~/.Tuxedo SSO/truststore.jks

   - Окна:

     c:\> kcadm config truststore --trustpass %ПАРОЛЬ% %HOMEPATH%\.Tuxedo SSO\truststore.jks

Конфиденциальные параметры

Конфиденциальные значения, такие как пароли, могут быть указаны как параметры команды. Обычно это не рекомендуется. Существуют также механизмы, с помощью которых можно запросить конфиденциальное значение — либо пропустив параметр, либо указав значение или -. Наконец, все будут иметь соответствующую переменную env, которую можно использовать вместо этого — проверьте справку по команде, которую вы запускаете, чтобы увидеть все возможные параметры.

Аутентификация

При входе в систему с помощью интерфейса командной строки администратора вы указываете:

- URL конечной точки сервера
- Царство
- Имя пользователя

Другой вариант — указать только clientId, что создаст для вас уникальную учетную запись службы.

При входе с использованием имени пользователя используйте пароль для указанного пользователя. При входе с использованием clientId вам нужен только секрет клиента, а не пароль пользователя. Вы также можете использовать Signed JWTвместо секрета клиента.

Убедитесь, что учетная запись, используемая для сеанса, имеет соответствующие разрешения для вызова операций Admin REST API. Например, realm-adminроль клиента realm-managementможет администрировать область пользователя.

Для аутентификации доступны два основных механизма. Один механизм используется kcadm config credentialsдля запуска аутентифицированного сеанса.

$ kcadm.sh учетные данные конфигурации --server http://localhost:8080 --realm master --user admin

Этот механизм поддерживает аутентифицированный сеанс между kcadmвызовами команд, сохраняя полученный токен доступа и связанный с ним токен обновления.

Он может поддерживать другие секреты в частном файле конфигурации. Подробнее см. в следующей главе .

Второй механизм аутентифицирует каждый вызов команды на время вызова. Этот механизм увеличивает нагрузку на сервер и время, затрачиваемое на круговые обходы для получения токенов. Преимущество этого подхода в том, что нет необходимости сохранять токены между вызовами, поэтому на диск ничего не сохраняется. Tuxedo SSO использует этот режим, когда --no-configуказан аргумент.

Например, при выполнении операции укажите всю информацию, необходимую для аутентификации.

$ kcadm.sh получить области --no-config --server http://localhost:8080 --realm master --user admin

Выполните kcadm.sh helpкоманду для получения дополнительной информации об использовании интерфейса командной строки администратора.

Выполните kcadm.sh config credentials --helpкоманду для получения дополнительной информации о запуске сеанса аутентификации.

Если вы не укажете параметр --password (обычно рекомендуется не указывать пароли в команде), вам будет предложено ввести пароль, если только он не указан в переменной среды KC_CLI_PASSWORD.

Работа с альтернативными конфигурациями

По умолчанию Admin CLI поддерживает файл конфигурации с именем kcadm.config. Tuxedo SSO помещает этот файл в домашний каталог пользователя. В системах на базе Linux полный путь — $HOME/.Tuxedo SSO/kcadm.config. В Windows полный путь — %HOMEPATH%\.Tuxedo SSO\ kcadm.config.

Вы можете использовать эту --configопцию, чтобы указать другой файл или местоположение, что позволит вам поддерживать несколько аутентифицированных сеансов параллельно.

Выполнение операций, связанных с одним файлом конфигурации, из одного потока.

Убедитесь, что файл конфигурации невидим для других пользователей в системе. Он содержит токены доступа и секреты, которые должны быть закрытыми. Tuxedo SSO ~/.Tuxedo SSOавтоматически создает каталог и его содержимое с надлежащими ограничениями доступа. Если каталог уже существует, Tuxedo SSO не обновляет разрешения каталога.

Можно избежать хранения секретов внутри файла конфигурации, но это неудобно и увеличивает количество запросов токенов. Используйте --no-configопцию со всеми командами и укажите информацию аутентификации, config credentialsкоторую команда требует при каждом вызове kcadm.

Базовые операции и URI ресурсов

Интерфейс командной строки администратора может выполнять операции CRUD в отношении конечных точек API REST администратора с помощью дополнительных команд, упрощающих выполнение определенных задач.

Основная схема использования приведена ниже:

$ kcadm.sh создать КОНЕЧНУЮ ТОЧКУ [АРГУМЕНТЫ]
$ kcadm.sh получить КОНЕЧНУЮ ТОЧКУ [АРГУМЕНТЫ]
$ kcadm.sh обновить КОНЕЧНУЮ ТОЧКУ [АРГУМЕНТЫ]
$ kcadm.sh удалить КОНЕЧНУЮ ТОЧКУ [АРГУМЕНТЫ]

Команды create, get, updateи deleteсоответствуют HTTP-глаголам POST, GET, PUT, и DELETE, соответственно. ENDPOINT — это целевой ресурсный URI, который может быть абсолютным (начинающимся с http:или https:) или относительным, который Tuxedo SSO использует для составления абсолютных URL-адресов в следующем формате:

SERVER_URI/admin/realms/REALM/КОНЕЧНАЯ ТОЧКА

Например, если вы выполняете аутентификацию на сервере http://localhost:8080 , a realm — master, использование usersв качестве ENDPOINT создает URL-адрес ресурса http://localhost:8080/admin/realms/master/users .

Если задать ENDPOINT равным clients, то эффективным URI ресурса будет http://localhost:8080/admin/realms/master/clients .

Tuxedo SSO имеет realmsконечную точку, которая является контейнером для областей. Она разрешается в:

SERVER_URI/admin/realms

Tuxedo SSO имеет serverinfoконечную точку. Эта конечная точка независима от областей.

При аутентификации в качестве пользователя с полномочиями администратора области вам может потребоваться выполнить команды в нескольких областях. Если это так, укажите опцию, -rчтобы явно указать CLI, в какой области должна выполняться команда. Вместо использования, REALMкак указано опцией --realm, kcadm.sh config credentialskоманда использует TARGET_REALM.

SERVER_URI/admin/realms/TARGET_REALM/ENDPOINT

Например:

$ kcadm.sh учетные данные конфигурации --server http://localhost:8080 --realm master --user admin
$ kcadm.sh создать пользователей -s имя_пользователя=testuser -s enabled=true -r demorealm

В этом примере вы начинаете сеанс, аутентифицированный как adminпользователь в masterобласти. Затем вы выполняете вызов POST по URL-адресу ресурса http://localhost:8080/admin/realms/demorealm/users.

Команды createи updateотправляют тело JSON на сервер. Вы можете использовать -f FILENAMEдля чтения готового документа из файла. Когда вы можете использовать опцию -f -, Tuxedo SSO считывает тело сообщения из стандартного ввода. Вы можете указать отдельные атрибуты и их значения, как показано в create usersпримере. Tuxedo SSO объединяет атрибуты в тело JSON и отправляет их на сервер.

> Значение в парах имя=значение, используемых в опциях --set, -s, предполагается JSON. Если его невозможно проанализировать как допустимый JSON, то оно будет отправлено на сервер как текстовое значение.
>
> Если значение заключено в кавычки после обработки оболочки, но не является допустимым JSON, кавычки будут удалены, а остальная часть значения будет отправлена как текст. Такое

поведение устарело, пожалуйста, рассмотрите возможность указания вашего значения без кавычек или допустимого строкового литерала JSON с двойными кавычками.

В Tuxedo SSO доступно несколько методов обновления ресурса с помощью updateкоманды. Вы можете определить текущее состояние ресурса и сохранить его в файл, отредактировать этот файл и отправить его на сервер для обновления.

Например:

```
$ kcadm.sh получить realms/demorealm > demorealm.json
$ vi demorealm.json
$ kcadm.sh обновить realms/demorealm -f demorealm.json
```

Этот метод обновляет ресурс на сервере с использованием атрибутов в отправленном документе JSON.

Другой метод — выполнить обновление «на лету», используя параметры -s, --setдля установки новых значений.

Например:

```
$ kcadm.sh обновить области/demorealm -s включено=false
```

Этот метод устанавливает enabledатрибут на false.

По умолчанию updateкоманда выполняет команду get, а затем объединяет новые значения атрибутов с существующими значениями. В некоторых случаях конечная точка может поддерживать putкоманду, но не getкоманду. Вы можете использовать -nопцию для выполнения обновления без слияния, которое выполняет putкоманду без предварительного запуска getкоманды.

Операции в сфере

Создание нового мира

Используйте createкоманду на realmsконечной точке для создания новой включенной области. Установите атрибуты на realmи enabled.

```
$ kcadm.sh создать области -s realm=demorealm -s enabled=true
```

Tuxedo SSO отключает области по умолчанию. Вы можете использовать область немедленно для аутентификации, включив ее.

Описание нового объекта также может быть в формате JSON.

$ kcadm.sh создать области -f demorealm.json

Вы можете отправить документ JSON с атрибутами области непосредственно из файла или передать документ на стандартный ввод.

Например:

- Линукс:

  $ kcadm.sh создать области -f - << EOF
  { "realm": "demorealm", "enabled": true }
  ЭОФ

- Окна:

  c:\> echo { "realm": "demorealm", "enabled": true } | kcadm создать области -f -

Список существующих областей

Эта команда возвращает список всех областей.

$ kcadm.sh получить области

> Tuxedo SSO фильтрует список областей на сервере, чтобы отображать только те области, которые может видеть пользователь.

Список всех атрибутов области может быть подробным, и большинство пользователей интересуются подмножеством атрибутов, таких как имя области и включенный статус области. Вы можете указать атрибуты для возврата, используя опцию --fields.

$ kcadm.sh получить области --fields область,включено

Результат можно отобразить в виде значений, разделенных запятыми.

$ kcadm.sh получить области --fields область --format csv --noquotes

Получение определенной области

Добавьте имя области к URI коллекции, чтобы получить отдельную область.

$ kcadm.sh получить области/мастер

Обновление области

1. Используйте эту -sвозможность, чтобы задать новые значения атрибутов, если вы не хотите менять все атрибуты области.

   Например:

   $ kcadm.sh обновить области/demorealm -s включено=false

2. Если вы хотите установить все записываемые атрибуты на новые значения:

   a. Выполните getкоманду.

   b. Отредактируйте текущие значения в файле JSON.

   c. Отправить повторно.

      Например:

      $ kcadm.sh получить realms/demorealm > demorealm.json
      $ vi demorealm.json
      $ kcadm.sh обновить realms/demorealm -f demorealm.json

Удаление области

Чтобы удалить область, выполните следующую команду:

$ kcadm.sh удалить области/демореалм

Включение всех опций страницы входа для области

Установите атрибуты, которые управляют определенными возможностями, на true.

Например:

$ kcadm.sh update realms/demorealm -s registrationAllowed=true -s registrationEmailAsUsername=true -s rememberMe=true -s verifyEmail=true -s resetPasswordAllowed=true -s editUsernameAllowed=true

Список ключей области

Используйте getоперацию на keysконечной точке целевой области.

$ kcadm.sh получить ключи -r demorealm

Генерация новых ключей области

1. Получите идентификатор целевой области перед добавлением новой пары ключей, сгенерированной RSA.

   Например:

   $ kcadm.sh получить области/демореальные области --fields id --format csv --noquotes

2. Добавьте нового поставщика ключей с более высоким приоритетом, чем у существующих поставщиков, как показано в kcadm.sh get keys -r demorealm.

   Например:

   - Линукс:

     $ kcadm.sh создать компоненты -r demorealm -s имя=rsa-generated -s providerId=rsa-generated -s providerType=org.Tuxedo SSO.keys.KeyProvider -s parentId=959844c1-d149-41d7-8359-6aa527fca0b0 -s 'config.priority=["101"]' -s 'config.enabled=["true"]' -s 'config.active=["true"]' -s 'config.keySize=["2048"]'

   - Окна:

     c:\> kcadm create components -r demorealm -s name=rsa-generated -s providerId=rsa-generated -s providerType=org.Tuxedo SSO.keys.KeyProvider -s parentId=959844c1-d149-41d7-8359-6aa527fca0b0 -s "config.priority=[\"101\"]" -s "config.enabled=[\"true\"]" -s "config.active=[\"true\"]" -s "config.keySize=[\"2048\"]"

3. Установите parentIdатрибут на значение идентификатора целевой области.

   Недавно добавленный ключ теперь является активным ключом, как показано в kcadm.sh get keys -r demorealm.

Добавление новых ключей области из файла хранилища ключей Java

1. Добавьте нового поставщика ключей, чтобы добавить новую пару ключей, заранее подготовленную в виде файла JKS.

   Например, на:

   - Линукс:

     $ kcadm.sh создает компоненты -r demorealm -s name=java-keystore -s providerId=java-keystore -s providerType=org.Tuxedo SSO.keys.KeyProvider -s parentId=959844c1-d149-41d7-8359-6aa527fca0b0 -s 'config.priority=["101"]' -s 'config.enabled=["true"]' -s 'config.active=["true"]' -s 'config.keystore=["/opt/Tuxedo SSO/keystore.jks"]' -s 'config.keystorePassword=["secret"]' -s 'config.keyPassword=["secret"]' -s 'config.keyAlias=["localhost"]'

   - Окна:

     c:\> kcadm create components -r demorealm -s name=java-keystore -s providerId=java-keystore -s providerType=org.Tuxedo SSO.keys.KeyProvider -s parentId=959844c1-d149-41d7-8359-6aa527fca0b0 -s "config.priority=[\"101\"]" -s "config.enabled=[\"true\"]" -s "config.active=[\"true\"]" -s "config.keystore=[\"/opt/Tuxedo SSO/keystore.jks\"]" -s "config.keystorePassword=[\"secret\"]" -s "config.keyPassword=[\"secret\"]" -s "config.keyAlias=[\"localhost\"]"

2. Обязательно измените значения
   атрибутов keystore, keystorePassword, keyPasswordи aliasв соответствии с вашим конкретным хранилищем ключей.
3. Установите parentIdатрибут на значение идентификатора целевой области.

Сделать ключ пассивным или отключить ключ

1. Определите клавишу, которую вы хотите сделать пассивной.

   $ kcadm.sh получить ключи -r demorealm

2. Используйте атрибут ключа providerIdдля создания URI конечной точки, например components/PROVIDER_ID.

3. Выполните update.

   Например:

   - Линукс:

     $ kcadm.sh обновить компоненты/PROVIDER_ID -r demorealm -s 'config.active=["false"]'

   - Окна:

     c:\> kcadm обновить компоненты/PROVIDER_ID -r demorealm -s "config.active=[\"false\"]"

     Вы можете обновить другие ключевые атрибуты:

   - Установите новое enabledзначение, чтобы отключить клавишу, например, config.enabled=["false"].
   - Установите новое priorityзначение, чтобы изменить приоритет ключа, например, config.priority=["110"].

Удаление старого ключа

1. Убедитесь, что удаляемый вами ключ неактивен и вы отключили его. Это действие предотвращает сбой существующих токенов, удерживаемых приложениями и пользователями.
2. Определите ключ, который необходимо удалить.

   $ kcadm.sh получить ключи -r demorealm

3. Используйте providerIdклавишу для выполнения удаления.

   $ kcadm.sh удалить компоненты/PROVIDER_ID -r demorealm

Настройка регистрации событий для области

Используйте updateкоманду на events/configконечной точке.

Атрибут eventsListenerscодержит список идентификаторов EventListenerProviderFactory, указывающих все прослушиватели событий, которые получают события. Доступны атрибуты, которые управляют встроенным

хранилищем событий, поэтому вы можете запрашивать прошлые события с помощью Admin REST API. Tuxedo SSO имеет отдельный контроль над регистрацией вызовов служб (eventsEnabled) и событиями аудита, инициированными Admin Console или Admin REST API (adminEventsEnabled). Вы можете настроить eventsExpirationсобытие на истечение срока действия, чтобы предотвратить заполнение базы данных. Tuxedo SSO устанавливает eventsExpirationвремя жизни, выраженное в секундах.

Вы можете настроить встроенный прослушиватель событий, который получает все события и регистрирует их через JBoss-logging. Используя регистратор org.Tuxedo SSO.events, Tuxedo SSO регистрирует события ошибок как , WARNа другие события как DEBUG.

Например:

- Линукс:

  $ kcadm.sh обновить события/конфигурацию -r demorealm -s 'eventsListeners=["jboss-logging"]'

- Окна:

  c:\> kcadm обновить события/конфигурацию -r demorealm -s "eventsListeners=[\"jboss-logging\"]"

Например:

Вы можете включить хранение всех доступных событий ERROR, за исключением событий аудита, на два дня, чтобы иметь возможность извлекать события через Admin REST.

- Линукс:

  $ kcadm.sh обновить события/конфигурацию -r demorealm -s eventsEnabled=true -s 'enabledEventTypes=["LOGIN_ERROR","REGISTER_ERROR","LOGOUT_ERROR","CODE_TO_TOKEN_ERROR","CLIENT_LOGIN_ERROR","FEDERATED_IDENTITY_LINK_ERROR","REMOVE_FEDERATED_IDENTITY_ERROR","UPDATE_EMAIL_ERROR","UPDATE_PROFILE_ERROR","UPDATE_PASSWORD_ERROR","UPDATE_TOTP_ERROR","UPDATE_CREDENTIAL_E

RROR","VERIFY_EMAIL_ERROR","REMOVE_TOTP_ERROR","REM
OVE_CREDENTIAL_ERROR", "SEND_VERIFY_EMAIL_ERROR",
"SEND_RESET_PASSWORD_ERROR",
"SEND_IDENTITY_PROVIDER_LINK_ERROR",
"RESET_PASSWORD_ERROR",
"IDENTITY_PROVIDER_FIRST_LOGIN_ERROR",
"IDENTITY_PROVIDER_POST_LOGIN_ERROR",
"CUSTOM_REQUIRED_ACTION_ERROR",
"EXECUTE_ACTIONS_ERROR", "CLIENT_REGISTER_ERROR",
"CLIENT_UPDATE_ERROR", "CLIENT_DELETE_ERROR"]' -s
eventsExpiration=172800

- Окна:

  c:\> kcadm update events/config -r demorealm -s eventsEnabled=true -s
  "enabledEventTypes=[\"LOGIN_ERROR\",\"REGISTER_ERROR\",\"LOGOUT
  _ERROR\",\"CODE_TO_TOKEN_ERROR\",\"CLIENT_LOGIN_ERROR\",\"F
  EDERATED_IDENTITY_LINK_ERROR\",\"REMOVE_FEDERATED_IDENTI
  TY_ERROR\",\"UPDATE_EMAIL_ERROR\",\"UPDATE_PROFILE_ERROR\",
  \"UPDATE_PASSWORD_ERROR\",\"UPDATE_TOTP_ERROR\",\"UPDATE_
  CREDENTIAL_ERROR\",\"VERIFY_EMAIL_ERROR\",\"REMOVE_TOTP_E
  RROR\",\"
  REMOVE_CREDENTIAL_ERROR\",\"SEND_VERIFY_EMAIL_ERROR\",\"S
  END_RESET_PASSWORD_ERROR\",\"SEND_IDENTITY_PROVIDER_LIN
  K_ERROR\",\"RESET_PASSWORD_ERROR\",\"IDENTITY_PROVIDER_FIR
  ST_LOGIN_ERROR\",\"IDENTITY_PROVIDER_POST_LOGIN_ERROR\",\"C
  USTOM_REQUIRED_ACTION_ERROR\",\"EXECUTE_ACTIONS_ERROR\",
  \"CLIENT_REGISTER_ERROR\",\"CLIENT_UPDATE_ERROR\",\"CLIENT_
  DELETE_ERROR\"]" -s eventsExpiration=172800

Вы можете сбросить сохраненные типы событий на все доступные типы событий . Установка значения в пустой список эквивалентна перечислению всех.

$ kcadm.sh обновить события/конфигурацию -r demorealm -s enabledEventTypes=[]

Вы можете включить хранение событий аудита.

$ kcadm.sh обновить события/конфигурацию -r demorealm -s
adminEventsEnabled=true -s adminEventsDetailsEnabled=true

Вы можете получить последние 100 событий. События упорядочены от самых новых к самым старым.

$ kcadm.sh получить события --offset 0 --limit 100

Вы можете удалить все сохраненные события.

$ kcadm удалить события

Очистка кэшей

1. Для очистки кэшей используйте createкоманду с одной из этих конечных точек:
    - clear-realm-cache
    - clear-user-cache
    - clear-keys-cache
2. Установите realmто же значение, что и у целевой области.

    Например:

    $ kcadm.sh создать очистку кэша-realm -r demorealm -s realm=demorealm
    $ kcadm.sh создать очистку кэша пользователя -r demorealm -s realm=demorealm
    $ kcadm.sh создать кэш-очистку-ключей -r demorealm -s realm=demorealm

Импорт области из экспортированного файла .json

1. Используйте createкоманду на partialImportконечной точке.
2. Установите ifResourceExistsзначение FAIL, SKIP, или OVERWRITE.
3. Используйте -fдля отправки экспортированного .jsonфайла области.

    Например:

    $ kcadm.sh создать частичный импорт -r demorealm2 -s ifResourceExists=FAIL -o -f demorealm.json

    Если область еще не существует, сначала создайте ее.

Например:

$ kcadm.sh создать области -s realm=demorealm2 -s enabled=true

Ролевые операции

Создание роли области

Используйте rolesконечную точку для создания роли области.

$ kcadm.sh create roles -r demorealm -s name=user -s 'description=Обычный пользователь с ограниченным набором разрешений'

Создание роли клиента

1. Определите клиента.
2. Используйте getкоманду для вывода списка доступных клиентов.

   $ kcadm.sh получить клиентов -r demorealm --fields id,clientId

3. Создайте новую роль, используя clientIdатрибут для построения URI конечной точки, например clients/ID/roles.

   Например:

   $ kcadm.sh create clients/a95b6af3-0bdc-4878-ae2e-6d61a4eca9a0/roles -r demorealm -s name=editor -s 'description=Редактор может редактировать и публиковать любую статью'

Список ролей области

Используйте getкоманду на rolesконечной точке, чтобы вывести список существующих ролей области.

$ kcadm.sh получить роли -r demorealm

Вы также можете использовать get-roleskоманду.

$ kcadm.sh получить-роли -r демореальм

Список ролей клиентов

Tuxedo SSO имеет специальную get-rolesкоманду для упрощения листинга ролей области и клиента. Команда является расширением команды getи ведет себя так же, как getкоманда, но с дополнительной семантикой для листинга ролей.

Используйте get-rolesкоманду, передав ей --cclientidпараметр clientId () или параметр id(--cid), чтобы идентифицировать клиента и вывести список ролей клиента.

Например:

$ kcadm.sh get-roles -r demorealm --cclientid realm-management

Получение определенной роли в сфере

Используйте getкоманду и роль nameдля создания URI конечной точки для определенной роли области, roles/ROLE_NAMEгде user— имя существующей роли.

Например:

$ kcadm.sh получить роли/пользователя -r demorealm

Вы можете использовать get-rolesкоманду, передав ей имя роли (--rolenameопция) или идентификатор (--roleidопция).

Например:

$ kcadm.sh get-roles -r demorealm --rolename пользователь

Получение определенной роли клиента

Используйте get-rolesкоманду, передав ей атрибут clientId (--cclientidопция) или атрибут ID (--cidопция) для идентификации клиента, и передайте имя роли (--rolenameопция) или атрибут ID роли (--roleid) для идентификации конкретной роли клиента.

Например:

$ kcadm.sh get-roles -r demorealm --cclientid realm-management --rolename manage-clients

Обновление роли области

Используйте updateкоманду с URI конечной точки, который вы использовали для получения определенной роли области.

Например:

$ kcadm.sh update roles/user -r demorealm -s 'description=Роль, представляющая обычного пользователя'

Обновление роли клиента

Используйте updateкоманду с URI конечной точки, которую вы использовали для получения определенной роли клиента.

Например:

$ kcadm.sh update clients/a95b6af3-0bdc-4878-ae2e-6d61a4eca9a0/roles/editor -r demorealm -s 'description=Пользователь, который может редактировать и публиковать статьи'

Удаление роли области

Используйте deleteкоманду с URI конечной точки, которую вы использовали для получения определенной роли области.

Например:

$ kcadm.sh удалить роли/пользователя -r demorealm

Удаление роли клиента

Используйте deleteкоманду с URI конечной точки, которую вы использовали для получения определенной роли клиента.

Например:

$ kcadm.sh удалить клиенты/a95b6af3-0bdc-4878-ae2e-6d61a4eca9a0/роли/редактор -r demorealm

Перечисление назначенных, доступных и эффективных ролей области для составной роли

Используйте эту get-rolesкоманду для вывода списка назначенных, доступных и эффективных ролей области для составной роли.

1. Чтобы составить список назначенных ролей области для составной роли, укажите целевую составную роль по имени (--rnameопция) или идентификатору (--ridопция).

   Например:

   $ kcadm.sh get-roles -r demorealm --rname testrole

2. Используйте эту --effectiveопцию для вывода списка действующих ролей области.

   Например:

   $ kcadm.sh get-roles -r demorealm --rname testrole --effective

3. Используйте эту --availableопцию для составления списка ролей области, которые можно добавить в составную роль.

   Например:

   $ kcadm.sh get-roles -r demorealm --rname testrole --available

Перечисление назначенных, доступных и эффективных клиентских ролей для составной роли

Используйте эту get-rolesкоманду для вывода списка назначенных, доступных и эффективных клиентских ролей для составной роли.

1. Чтобы вывести список назначенных клиентских ролей для составной роли, можно указать целевую составную роль по имени (--rnameопция) или идентификатору (--ridопция), а клиента — по атрибуту clientId (--cclientidопция) или идентификатору (--cidопция).

   Например:

   $ kcadm.sh get-roles -r demorealm --rname testrole --cclientid realm-management

2. Используйте эту --effectiveопцию для вывода списка действующих ролей области.

   Например:

   $ kcadm.sh get-roles -r demorealm --rname testrole --cclientid realm-management --effective

3. Используйте эту --availableопцию для вывода списка ролей области, которые можно добавить к целевой составной роли.

   Например:

   $ kcadm.sh get-roles -r demorealm --rname testrole --cclientid realm-management --available

Добавление ролей области к составной роли

Tuxedo SSO предоставляет add-roleskоманду для добавления ролей области и ролей клиента.

В этом примере роль добавляется userк составной роли testrole.

$ kcadm.sh add-roles --rname testrole --rolename user -r demorealm

Удаление ролей области из составной роли

Tuxedo SSO предоставляет remove-roleskоманду для удаления ролей области и ролей клиента.

В следующем примере удаляется userроль из целевой составной роли testrole.

$ kcadm.sh remove-roles --rname testrole --rolename user -r demorealm

Добавление клиентских ролей в роль области

Tuxedo SSO предоставляет add-roleskоманду для добавления ролей области и ролей клиента.

В следующем примере роли, определенные на клиенте, и , добавляются realm-managementв create-clientсоставную view-usersроль testrole.

$ kcadm.sh add-roles -r demorealm --rname testrole --cclientid realm-management --rolename create-client --rolename view-users

Добавление клиентских ролей к клиентской роли

1. Определите идентификатор составной клиентской роли с помощью get-roleskоманды.

   Например:

   $ kcadm.sh get-roles -r demorealm --cclientid test-client --rolename операции

2. Предположим, что существует клиент с атрибутом clientId с именем test-client, ролью клиента с именем supportи ролью клиента с именем , operationskоторая становится составной ролью с идентификатором «fc400897-ef6a-4e8c-872b-1581b7fa8a71».
3. Используйте следующий пример для добавления еще одной роли к составной роли.

   $ kcadm.sh add-roles -r demorealm --cclientid тестовый-клиент --rid fc400897-ef6a-4e8c-872b-1581b7fa8a71 --rolename поддержка

4. Выведите список ролей составной роли с помощью get-roles --allкоманды.

   Например:

   $ kcadm.sh получить-роли --rid fc400897-ef6a-4e8c-872b-1581b7fa8a71 --all

Удаление клиентских ролей из составной роли

Используйте remove-roleskоманду для удаления клиентских ролей из составной роли.

Используйте следующий пример для удаления двух ролей, определенных на клиенте realm-management, create-clientроли и view-usersроли, из testrolecоставной роли.

$ kcadm.sh remove-roles -r demorealm --rname testrole --cclientid realm-management --rolename create-client --rolename view-users

Добавление ролей клиентов в группу

Используйте add-rolesкоманду для добавления ролей области и ролей клиента.

Следующий пример добавляет роли, определенные на клиенте realm-management, create-clientи view-users, в Groupгруппу (--gnameопция). В качестве альтернативы вы можете указать группу по идентификатору (--gidопция).

Более подробную информацию см. в разделе «Групповые операции» .

$ kcadm.sh add-roles -r demorealm --gname Группа --cclientid realm-management --rolename создать-клиента --rolename просмотр-пользователей

Удаление клиентских ролей из группы

Используйте remove-rolesкоманду для удаления клиентских ролей из группы.

В следующем примере удаляются две роли, определенные на клиенте realm management, create-clientи view-users, из Groupгруппы.

Более подробную информацию см. в разделе «Групповые операции» .

$ kcadm.sh remove-roles -r demorealm --gname Группа --cclientid realm-management --rolename создать-клиента --rolename просмотр-пользователей

Клиентские операции

Создание клиента

1. Запустите createкоманду на clientsконечной точке, чтобы создать нового клиента.

   Например:

   $ kcadm.sh создать клиентов -r demorealm -s clientId=myapp -s enabled=true

2. Укажите секрет, если необходимо задать секрет для аутентификации адаптеров.

   Например:

```
$ kcadm.sh создать клиентов -r demorealm -s clientId=myapp -s enabled=true -s
clientAuthenticatorType=client-secret -s secret=d0b8122f-8dfb-46b7-b68a-
f5cc4e25d000
```

Список клиентов

Используйте getкоманду на clientsконечной точке для вывода списка клиентов.

В этом примере вывод фильтруется для отображения только атрибутов idи clientId:

```
$ kcadm.sh получить клиентов -r demorealm --fields id,clientId
```

Получение конкретного клиента

Используйте идентификатор клиента для создания URI конечной точки, нацеленной на конкретного клиента, например clients/ID.

Например:

```
$ kcadm.sh получить клиенты/c7b8547f-e748-4333-95d0-410b76b3f4a3 -r demorealm
```

Получение текущего секрета для конкретного клиента

Используйте идентификатор клиента для создания URI конечной точки, например clients/ID/client-secret.

Например:

```
$ kcadm.sh получить клиенты/$CID/client-secret -r demorealm
```

Сгенерировать новый секрет для конкретного клиента

Используйте идентификатор клиента для создания URI конечной точки, например clients/ID/client-secret.

Например:

```
$ kcadm.sh создать клиенты/$CID/client-secret -r demorealm
```

Обновление текущего секрета для конкретного клиента

Используйте идентификатор клиента для создания URI конечной точки, например clients/ID.

Например:

$ kcadm.sh обновить клиенты/$CID -s "secret=newSecret" -r demorealm

Получение файла конфигурации адаптера (Tuxedo SSO.json) для конкретного клиента

Используйте идентификатор клиента для создания URI конечной точки, нацеленной на конкретного клиента,
например clients/ID/installation/providers/Tuxedo SSO-oidc-Tuxedo SSO-json.

Например:

$ kcadm.sh получить клиенты/c7b8547f-e748-4333-95d0-410b76b3f4a3/installation/providers/Tuxedo SSO-oidc-Tuxedo SSO-json -r demorealm

Получение конфигурации адаптера подсистемы WildFly для конкретного клиента

Используйте идентификатор клиента для создания URI конечной точки, нацеленной на конкретного клиента,
например clients/ID/installation/providers/Tuxedo SSO-oidc-jboss-subsystem.

Например:

$ kcadm.sh получить клиенты/c7b8547f-e748-4333-95d0-410b76b3f4a3/installation/providers/Tuxedo SSO-oidc-jboss-subsystem -r demorealm

Получение примера конфигурации Docker-v2 для конкретного клиента

Используйте идентификатор клиента для создания URI конечной точки, нацеленной на конкретного клиента,
например clients/ID/installation/providers/docker-v2-compose-yaml.

Ответ в .zipформате.

Например:

$ kcadm.sh получить http://localhost:8080/admin/realms/demorealm/clients/8f271c35-44e3-446f-8953-b0893810ebe7/installation/providers/docker-v2-compose-yaml -r demorealm > Tuxedo SSO-docker-compose-yaml.zip

Обновление клиента

Используйте updateкоманду с тем же URI конечной точки, который вы используете для получения конкретного клиента.

Например:

- Линукс:

  $ kcadm.sh обновить клиенты/c7b8547f-e748-4333-95d0-410b76b3f4a3 -r demorealm -s enabled=false -s publicClient=true -s 'redirectUris=["http://localhost:8080/myapp/*"]' -s baseUrl=http://localhost:8080/myapp -s adminUrl=http://localhost:8080/myapp

- Окна:

  c:\> kcadm обновить клиенты/c7b8547f-e748-4333-95d0-410b76b3f4a3 -r demorealm -s enabled=false -s publicClient=true -s "redirectUris=[\"http://localhost:8080/myapp/*\"]" -s baseUrl=http://localhost:8080/myapp -s adminUrl=http://localhost:8080/myapp

Удаление клиента

Используйте deleteкоманду с тем же URI конечной точки, который вы используете для получения конкретного клиента.

Например:

$ kcadm.sh удалить клиенты/c7b8547f-e748-4333-95d0-410b76b3f4a3 -r demorealm

Добавление или удаление ролей для учетной записи клиента

Учетная запись клиента — это учетная запись пользователя с именем пользователя service-account-CLIENT_ID. Вы можете выполнять те же пользовательские операции на этой учетной записи, что и на обычной учетной записи.

Пользовательские операции

Создание пользователя

Запустите createкоманду на usersконечной точке, чтобы создать нового пользователя.

Например:

$ kcadm.sh создать пользователей -r demorealm -s имя_пользователя=testuser -s enabled=true

Список пользователей

Используйте usersконечную точку для составления списка пользователей. Целевой пользователь должен сменить свой пароль при следующем входе в систему.

Например:

$ kcadm.sh получить пользователей -r demorealm --offset 0 --limit 1000

Вы можете фильтровать пользователей по username, firstName, lastName, или email.

Например:

$ kcadm.sh получить пользователей -r demorealm -qq=email:google.com
$ kcadm.sh получить пользователей -r demorealm -qq=имя_пользователя:testuser

> Фильтрация не использует точное соответствие. Этот пример сопоставляет значение атрибута `username`с `*testuser*`шаблоном.

Для клиентов, групп и пользователей вы можете фильтровать по нескольким атрибутам, указав более сложный qпараметр запроса. Вы можете использовать что-то вроде -qq="field1:value1 field2:value2". Tuxedo SSO возвращает пользователей, которые соответствуют условию только для всех атрибутов.

Получение конкретного пользователя

Используйте идентификатор пользователя для создания URI конечной точки, например users/USER_ID.

Например:

$ kcadm.sh получить пользователей/0ba7a3fd-6fd8-48cd-a60b-2e8fd82d56e2 -r demorealm

Обновление пользователя

Используйте updateкоманду с тем же URI конечной точки, который вы используете для получения конкретного пользователя.

Например:

- Линукс:

  $ kcadm.sh обновить пользователей/0ba7a3fd-6fd8-48cd-a60b-2e8fd82d56e2 -r demorealm -s 'requiredActions=["VERIFY_EMAIL","UPDATE_PROFILE","CONFIGURE_TOTP","UPDATE_PASSWORD"]'

- Окна:

  c:\> kcadm update users/0ba7a3fd-6fd8-48cd-a60b-2e8fd82d56e2 -r demorealm -s "requiredActions=[\"VERIFY_EMAIL\",\"UPDATE_PROFILE\",\"CONFIGURE_TOTP\",\"UPDATE_PASSWORD\"]"

Удаление пользователя

Используйте deleteкоманду с тем же URI конечной точки, который вы используете для получения конкретного пользователя.

Например:

$ kcadm.sh удалить пользователей/0ba7a3fd-6fd8-48cd-a60b-2e8fd82d56e2 -r demorealm

Сброс пароля пользователя

Используйте специальную set-passwordкоманду для сброса пароля пользователя.

Например:

$ kcadm.sh set-password -r demorealm --username testuser --new-password НОВЫЙПАРОЛЬ --temporary

Эта команда устанавливает временный пароль для пользователя. Целевой пользователь должен сменить пароль при следующем входе в систему.

Вы можете --useridуказать пользователя с помощью idатрибута.

Того же результата можно добиться, используя updateкоманду на конечной точке, созданной на основе той, которую вы использовали для получения конкретного пользователя, например users/USER_ID/reset-password.

Например:

$ kcadm.sh обновить пользователей/0ba7a3fd-6fd8-48cd-a60b-2e8fd82d56e2/reset-password -r demorealm -s тип=пароль -s значение=НОВЫЙПАРОЛЬ -s временный=истина -n

Параметр -nгарантирует, что Tuxedo SSO выполнит PUTкоманду без выполнения GETкоманды перед PUTкомандой. Это необходимо, поскольку reset-passwordконечная точка не поддерживает GET.

Список назначенных, доступных и эффективных ролей области для пользователя

Вы можете использовать get-roleskоманду для вывода списка назначенных, доступных и эффективных ролей области для пользователя.

1. Укажите целевого пользователя по имени или идентификатору, чтобы вывести список назначенных пользователю ролей области.

   Например:

   $ kcadm.sh get-roles -r demorealm --uusername testuser

2. Используйте эту --effectiveопцию для вывода списка действующих ролей области.

   Например:

   $ kcadm.sh get-roles -r demorealm --uusername testuser --effective

3. Используйте эту --availableопцию для вывода списка ролей области, которые можно добавить пользователю.

   Например:

```
$ kcadm.sh get-roles -r demorealm --uusername testuser --available
```

Перечисление назначенных, доступных и эффективных клиентских ролей для пользователя

Используйте get-rolesкоманду для вывода списка назначенных, доступных и эффективных клиентских ролей для пользователя.

1. Укажите целевого пользователя по имени пользователя (--uusernameопция) или идентификатору (--uidопция), а клиента — по атрибуту clientId (--cclientidопция) или идентификатору (--cidопция), чтобы получить список назначенных клиентских ролей для пользователя.

   Например:

   ```
   $ kcadm.sh get-roles -r demorealm --uusername testuser --cclientid realm-management
   ```

2. Используйте эту --effectiveопцию для вывода списка действующих ролей области.

   Например:

   ```
   $ kcadm.sh get-roles -r demorealm --uusername testuser --cclientid realm-management --effective
   ```

3. Используйте эту --availableопцию для вывода списка ролей области, которые можно добавить пользователю.

   Например:

   ```
   $ kcadm.sh get-roles -r demorealm --uusername testuser --cclientid realm-management --available
   ```

Добавление ролей области пользователю

Используйте add-rolesкоманду для добавления ролей области пользователю.

Используйте следующий пример для добавления userроли пользователю testuser:

```
$ kcadm.sh add-roles --uusername testuser --rolename user -r demorealm
```

Удаление ролей области у пользователя

Используйте remove-rolesкоманду для удаления ролей области у пользователя.

userЧтобы удалить роль пользователя, используйте следующий пример testuser:

$ kcadm.sh remove-roles --uusername testuser --rolename user -r demorealm

Добавление клиентских ролей пользователю

Используйте add-rolesкоманду для добавления клиентских ролей пользователю.

Используйте следующий пример, чтобы добавить две роли, определенные на клиенте realm management, create-clientроль и view-usersроль, к пользователю testuser.

$ kcadm.sh add-roles -r demorealm --uusername testuser --cclientid realm-management --rolename create-client --rolename view-users

Удаление клиентских ролей у пользователя

Используйте remove-rolesкоманду для удаления клиентских ролей у пользователя.

Используйте следующий пример для удаления двух ролей, определенных в клиенте управления областью:

$ kcadm.sh remove-roles -r demorealm --uusername testuser --cclientid realm-management --rolename create-client --rolename view-users

Список сеансов пользователя

1. Определить идентификатор пользователя,
2. Используйте идентификатор для составления URI конечной точки, например users/ID/sessions.
3. Используйте getкоманду для получения списка сеансов пользователя.

   Например:

   $ kcadm.sh получить пользователи/6da5ab89-3397-4205-afaa-e201ff638f9e/sessions -r demorealm

Выход пользователя из определенного сеанса

1. Определите идентификатор сеанса, как описано ранее.
2. Используйте идентификатор сеанса для создания URI конечной точки, например sessions/ID.
3. Используйте deleteкоманду для отмены сеанса.

   Например:

   $ kcadm.sh удалить сеансы/d0eaa7cc-8c5d-489d-811a-69d3c4ec84d1 -r demorealm

Выход пользователя из всех сеансов

Используйте идентификатор пользователя для создания URI конечной точки, например users/ID/logout.

Используйте createкоманду для выполнения POSTна этом URI конечной точки.

Например:

$ kcadm.sh create users/6da5ab89-3397-4205-afaa-e201ff638f9e/logout -r demorealm -s realm=demorealm -s user=6da5ab89-3397-4205-afaa-e201ff638f9e

Групповые операции

Создание группы

Используйте createкоманду на groupsконечной точке для создания новой группы.

Например:

$ kcadm.sh создать группы -r demorealm -s имя=Группа

Листинг групп

Используйте getкоманду на groupsконечной точке для вывода списка групп.

Например:

$ kcadm.sh получить группы -r demorealm

Получение определенной группы

Используйте идентификатор группы для создания URI конечной точки, например groups/GROUP_ID.

Например:

$ kcadm.sh получить группы/51204821-0580-46db-8f2d-27106c6b5ded -r demorealm

Обновление группы

Используйте updateкоманду с тем же URI конечной точки, который вы используете для получения определенной группы.

Например:

$ kcadm.sh обновить groups/51204821-0580-46db-8f2d-27106c6b5ded -s 'attributes.email=["group@example.com"]' -r demorealm

Удаление группы

Используйте deleteкоманду с тем же URI конечной точки, который вы используете для получения определенной группы.

Например:

$ kcadm.sh удалить группы/51204821-0580-46db-8f2d-27106c6b5ded -r demorealm

Создание подгруппы

Найдите идентификатор родительской группы, перечислив группы. Используйте этот идентификатор для построения URI конечной точки, например groups/GROUP_ID/children.

Например:

$ kcadm.sh создать группы/51204821-0580-46db-8f2d-27106c6b5ded/children -r demorealm -s имя=Подгруппа

Перемещение группы под другую группу

1. Найдите идентификатор существующей родительской группы и идентификатор существующей дочерней группы.

2. Используйте идентификатор родительской группы для создания URI конечной точки, например groups/PARENT_GROUP_ID/children.
3. Запустите createкоманду на этой конечной точке и передайте идентификатор дочерней группы в виде тела JSON.

Например:

$ kcadm.sh создать группы/51204821-0580-46db-8f2d-27106c6b5ded/children -r demorealm -s id=08d410c6-d585-4059-bb07-54dcb92c5094 -s имя=Подгруппа

Получить группы для определенного пользователя

Используйте идентификатор пользователя для определения членства пользователя в группах для составления URI конечной точки, например users/USER_ID/groups.

Например:

$ kcadm.sh получить пользователи/b544f379-5fc4-49e5-8a8d-5cfb71f46f53/группы -r demorealm

Добавление пользователя в группу

Используйте updateкоманду с URI конечной точки, состоящим из идентификатора пользователя и идентификатора группы,
например users/USER_ID/groups/GROUP_ID, чтобы добавить пользователя в группу.

Например:

$ kcadm.sh обновить users/b544f379-5fc4-49e5-8a8d-5cfb71f46f53/groups/ce01117a-7426-4670-a29a-5c118056fe20 -r demorealm -s realm=demorealm -s userId=b544f379-5fc4-49e5-8a8d-5cfb71f46f53 -s groupId=ce01117a-7426-4670-a29a-5c118056fe20 -n

Удаление пользователя из группы

Чтобы удалить пользователя из группы, используйте deleteкоманду на том же URI конечной точки, который вы используете для добавления пользователя в группу, например ,.users/USER_ID/groups/GROUP_ID

Например:

$ kcadm.sh удалить пользователей/b544f379-5fc4-49e5-8a8d-5cfb71f46f53/groups/ce01117a-7426-4670-a29a-5c118056fe20 -r demorealm

Список назначенных, доступных и эффективных ролей области для группы

Используйте специальную get-rolesкоманду для вывода списка назначенных, доступных и эффективных ролей области для группы.

1. Укажите целевую группу по имени (--gnameопция), пути (--gpathопция) или идентификатору (--gidопция), чтобы вывести список назначенных ролей области для группы.

   Например:

   $ kcadm.sh get-roles -r demorealm --gname Группа

2. Используйте эту --effectiveопцию для вывода списка действующих ролей области.

   Например:

   $ kcadm.sh get-roles -r demorealm --gname Группа --effective

3. Используйте эту --availableопцию для вывода списка ролей области, которые вы можете добавить в группу.

   Например:

   $ kcadm.sh get-roles -r demorealm --gname Группа --available

Перечисление назначенных, доступных и эффективных ролей клиентов для группы

Используйте эту get-rolesкоманду для вывода списка назначенных, доступных и эффективных клиентских ролей для группы.

1. Укажите целевую группу по имени (--gnameопция) или идентификатору (--gidопция),
2. Укажите клиента по атрибуту clientId (--cclientidопция) или ID (--idопция), чтобы вывести список назначенных клиентских ролей для пользователя.

Например:

$ kcadm.sh get-roles -r demorealm --gname Группа --cclientid realm-management

3. Используйте эту --effectiveопцию для вывода списка действующих ролей области.

   Например:

   $ kcadm.sh get-roles -r demorealm --gname Группа --cclientid realm-management --effective

4. Используйте эту --availableопцию для вывода списка ролей области, которые вы все еще можете добавить в группу.

   Например:

   $ kcadm.sh get-roles -r demorealm --gname Группа --cclientid realm-management --available

Операции поставщика удостоверений

Список доступных поставщиков удостоверений

Используйте serverinfoконечную точку для получения списка доступных поставщиков удостоверений.

Например:

$ kcadm.sh получить информацию о сервере -r demorealm --fields 'identityProviders(*)'

> Tuxedo SSO обрабатывает `serverinfo`конечную точку аналогично `realms`конечной точке. Tuxedo SSO не разрешает конечную точку относительно целевой области, поскольку она существует вне какой-либо конкретной области.

Список настроенных поставщиков удостоверений

Используйте identity-provider/instancesконечную точку.

Например:

$ kcadm.sh получить идентификатор-провайдера/экземпляры -r demorealm --fields псевдоним,providerId,включено

Получение определенного настроенного поставщика удостоверений

Используйте атрибут поставщика удостоверений aliasдля создания URI конечной точки, например identity-provider/instances/ALIAS, , чтобы получить конкретного поставщика удостоверений.

Например:

$ kcadm.sh получить идентификатор-провайдера/экземпляры/facebook -r demorealm

Удаление определенного настроенного поставщика удостоверений

Используйте deleteкоманду с тем же URI конечной точки, который вы используете для получения определенного настроенного поставщика удостоверений, чтобы удалить определенного настроенного поставщика удостоверений.

Например:

$ kcadm.sh удалить поставщик-идентификации/экземпляры/facebook -r demorealm

Настройка поставщика удостоверений Tuxedo SSO OpenID Connect

1. Используйте его Tuxedo SSO-oidcпри providerIdсоздании нового экземпляра поставщика удостоверений.
2. Укажите configатрибуты: authorizationUrl, tokenUrl, clientId, и clientSecret.

   Например:

   $ kcadm.sh создать идентификацию-поставщика/экземпляров -r demorealm -s alias=Tuxedo SSO-oidc -s providerId=Tuxedo SSO-oidc -s enabled=true -s 'config.useJwksUrl="true"' -s config.authorizationUrl=http://localhost:8180/realms/demorealm/protocol/openid-connect/auth -s config.tokenUrl=http://localhost:8180/realms/demorealm/protocol/openid-connect/token -s config.clientId=demo-oidc-provider -s config.clientSecret=secret

Настройка поставщика удостоверений OpenID Connect

Настройте универсального поставщика OpenID Connect так же, как вы настраиваете поставщика Tuxedo SSO OpenID Connect, за исключением того, что вы устанавливаете providerIdзначение атрибута равным oidc.

Настройка поставщика удостоверений SAML 2

1. Использовать samlкак providerId.
2. Укажите configатрибуты: singleSignOnServiceUrl, nameIDPolicyFormat, и signatureAlgorithm.

Например:

$ kcadm.sh создать идентификационный-провайдер/экземпляры -r demorealm -s alias=saml -s providerId=saml -s enabled=true -s 'config.useJwksUrl="true"' -s config.singleSignOnServiceUrl=http://localhost:8180/realms/saml-broker-realm/protocol/saml -s config.nameIDPolicyFormat=urn:oasis:names:tc:SAML:2.0:nameid-format:persistent -s config.signatureAlgorithm=RSA_SHA256

Настройка поставщика удостоверений Facebook

1. Использовать facebookкак providerId.
2. Укажите configатрибуты: clientIdи clientSecret. Вы можете найти эти атрибуты на странице конфигурации приложения Facebook Developers для вашего приложения. Для получения дополнительной информации см. страницу брокера идентификации Facebook .

   Например:

   $ kcadm.sh создать поставщика удостоверений/экземпляры -r demorealm -s alias=facebook -s providerId=facebook -s enabled=true -s 'config.useJwksUrl="true"' -s config.clientId=FACEBOOK_CLIENT_ID -s config.clientSecret=FACEBOOK_CLIENT_SECRET

Настройка поставщика удостоверений Google

1. Использовать googlекак providerId.

2. Укажите configатрибуты: clientIdи clientSecret. Вы можете найти эти атрибуты на странице конфигурации приложения Google Developers для вашего приложения. Для получения дополнительной информации см. страницу брокера идентификации Google .

   Например:

   $ kcadm.sh создать поставщика удостоверений/экземпляры -r demorealm -s alias=google -s providerId=google -s enabled=true -s 'config.useJwksUrl="true"' -s config.clientId=GOOGLE_CLIENT_ID -s config.clientSecret=GOOGLE_CLIENT_SECRET

Настройка поставщика удостоверений Twitter

1. Использовать twitterкак providerId.
2. Укажите configатрибуты clientIdи clientSecret. Вы можете найти эти атрибуты на странице конфигурации приложения Twitter Application Management для вашего приложения. Для получения дополнительной информации см. страницу брокера идентификации Twitter .

   Например:

   $ kcadm.sh создать поставщика удостоверений/экземпляры -r demorealm -s alias=google -s providerId=google -s enabled=true -s 'config.useJwksUrl="true"' -s config.clientId=TWITTER_API_KEY -s config.clientSecret=TWITTER_API_SECRET

Настройка поставщика удостоверений GitHub

1. Использовать githubкак providerId.
2. Укажите configатрибуты clientIdи clientSecret. Вы можете найти эти атрибуты на странице настроек приложения разработчика GitHub для вашего приложения. Для получения дополнительной информации см. страницу брокера идентификации GitHub .

   Например:

   $ kcadm.sh создать поставщика удостоверений/экземпляры -r demorealm -s alias=github -s providerId=github -s enabled=true -s 'config.useJwksUrl="true"' -s

config.clientId=GITHUB_CLIENT_ID -s
config.clientSecret=GITHUB_CLIENT_SECRET

Настройка поставщика удостоверений LinkedIn

1. Использовать linkedinкак providerId.
2. Укажите configатрибуты clientIdи clientSecret. Вы можете найти эти атрибуты на странице приложения LinkedIn Developer Console для вашего приложения. Для получения дополнительной информации см. страницу брокера идентификации LinkedIn .

   Например:

   $ kcadm.sh создать поставщика удостоверений/экземпляры -r demorealm -s alias=linkedin -s providerId=linkedin -s enabled=true -s 'config.useJwksUrl="true"' -s config.clientId=LINKEDIN_CLIENT_ID -s config.clientSecret=LINKEDIN_CLIENT_SECRET

Настройка поставщика удостоверений Microsoft Live

1. Использовать microsoftкак providerId.
2. Укажите configатрибуты clientIdи clientSecret. Вы можете найти эти атрибуты на странице портала регистрации приложений Microsoft для вашего приложения. Для получения дополнительной информации см. страницу брокера идентификации Microsoft .

   Например:

   $ kcadm.sh создать поставщика удостоверений/экземпляры -r demorealm -s alias=microsoft -s providerId=microsoft -s enabled=true -s 'config.useJwksUrl="true"' -s config.clientId=MICROSOFT_APP_ID -s config.clientSecret=MICROSOFT_PASSWORD

Настройка поставщика удостоверений Stack Overflow

1. Используйте stackoverflowкоманду как providerId.
2. Укажите configатрибуты clientId, clientSecretи key. Вы можете найти эти атрибуты на странице Stack Apps OAuth для вашего приложения. Для

получения дополнительной информации см. страницу брокера идентификации Stack Overflow .

Например:

$ kcadm.sh создать поставщика удостоверений/экземпляры -r demorealm -s alias=stackoverflow -s providerId=stackoverflow -s enabled=true -s 'config.useJwksUrl="true"' -s config.clientId=STACKAPPS_CLIENT_ID -s config.clientSecret=STACKAPPS_CLIENT_SECRET -s config.key=STACKAPPS_KEY

Операции поставщика услуг хранения данных

Настройка поставщика хранилища Kerberos

1. Используйте createкоманду для componentsконечной точки.
2. Укажите идентификатор области в качестве значения атрибута parentId.
3. Укажите kerberosв качестве значения атрибута providerId, a org.Tuxedo SSO.storage.UserStorageProviderв качестве значения атрибута providerType.
4. Например:

   $ kcadm.sh создать компоненты -r demorealm -s parentId=demorealmId -s id=demokerberos -s name=demokerberos -s providerId=kerberos -s providerType=org.Tuxedo SSO.storage.UserStorageProvider -s 'config.priority=["0"]' -s 'config.debug=["false"]' -s 'config.allowPasswordAuthentication=["true"]' -s 'config.editMode=["UNSYNCED"]' -s 'config.updateProfileFirstLogin=["true"]' -s 'config.allowKerberosAuthentication=["true"]' -s 'config.kerberosRealm=["Tuxedo SSO.ORG"]' -s 'config.keyTab=["http.keytab"]' -s 'config.serverPrincipal=["HTTP/localhost@Tuxedo SSO.ORG"]' -s 'config.cachePolicy=["DEFAULT"]'

Настройка поставщика хранилища пользователей LDAP

1. Используйте createкоманду для componentsконечной точки.
2. Укажите ldapв качестве значения атрибута providerId, a org.Tuxedo SSO.storage.UserStorageProviderв качестве значения атрибута providerType.

3. Укажите идентификатор области в качестве значения атрибута parentId.
4. Используйте следующий пример для создания поставщика LDAP, интегрированного с Kerberos.

   $ kcadm.sh создать компоненты -r demorealm -s имя=kerberos-ldap-provider -s providerId=ldap -s providerType=org.Tuxedo SSO.storage.UserStorageProvider -s parentId=3d9c572b-8f33-483f-98a6-8bb421667867 -s 'config.priority=["1"]' -s 'config.fullSyncPeriod=["-1"]' -s 'config.changedSyncPeriod=["-1"]' -s 'config.cachePolicy=["DEFAULT"]' -s config.evictionDay=[] -s config.evictionHour=[] -s config.evictionMinute=[] -s config.maxLifespan=[] -s 'config.batchSizeForSync=["1000"]' -s 'config.editMode=["WRITABLE"]' -s 'config.syncRegistrations=["false"]' -s 'config.vendor=["other"]' -s 'config.usernameLDAPAttribute=["uid"]' -s 'config.rdnLDAPAttribute=["uid"]' -s 'config.uuidLDAPAttribute=["entryUUID"]' -s 'config.userObjectClasses=["inetOrgPerson, organizationalPerson"]' -s 'config.connectionUrl=["ldap://localhost:10389"]' -s 'config.usersDn=["ou=People,dc=Tuxedo SSO,dc=org"]' -s 'config.authType=["simple"]' -s 'config.bindDn=["uid=admin,ou=system"]' -s 'config.bindCredential=["secret"]' -s 'config.searchScope=["1"]' -s 'config.useTruststoreSpi=["always"]' -s 'config.connectionPooling=["true"]' -s 'config.pagination=["true"]' -s 'config.allowKerberosAuthentication=["true"]' -s 'config.serverPrincipal=["HTTP/localhost@Tuxedo SSO.ORG"]' -s 'config.keyTab=["http.keytab"]' -s 'config.kerberosRealm=["Tuxedo SSO.ORG"]' -s 'config.debug=["true"]' -s 'config.useKerberosForPasswordAuthentication=["true"]'

Удаление экземпляра поставщика хранилища пользователя

1. Используйте атрибут экземпляра поставщика хранилища idдля создания URI конечной точки, например components/ID.
2. Запустите deleteкоманду для этой конечной точки.

   Например:

   $ kcadm.sh удалить компоненты/3d9c572b-8f33-483f-98a6-8bb421667867 -r demorealm

Запуск синхронизации всех пользователей для определенного поставщика хранилища пользователей

1. Используйте атрибут поставщика хранилища idдля создания URI конечной точки, например user-storage/ID_OF_USER_STORAGE_INSTANCE/sync.
2. Добавьте action=triggerFullSyncпараметр запроса.
3. Выполните createкоманду.

   Например:

   $ kcadm.sh создать пользовательское-хранилище/b7c63d02-b62a-4fc1-977c-947d6a09e1ea/sync?action=triggerFullSync

Запуск синхронизации измененных пользователей для определенного поставщика хранилища пользователей

1. Используйте атрибут поставщика хранилища idдля создания URI конечной точки, например user-storage/ID_OF_USER_STORAGE_INSTANCE/sync.
2. Добавьте action=triggerChangedUsersSyncпараметр запроса.
3. Выполните createкоманду.

   Например:

   $ kcadm.sh создать пользовательское-хранилище/b7c63d02-b62a-4fc1-977c-947d6a09e1ea/sync?action=triggerChangedUsersSync

Тестовое подключение к хранилищу пользователя LDAP

1. Запустите getкоманду на testLDAPConnectionконечной точке.
2. Укажите параметры запроса bindCredential, bindDn, connectionUrl, и useTruststoreSpi.
3. Установите actionпараметр запроса на testConnection.

   Например:

   $ kcadm.sh create testLDAPConnection -s action=testConnection -s bindCredential=secret -s bindDn=uid=admin,ou=system -s connectionUrl=ldap://localhost:10389 -s useTruststoreSpi=always

Тестовая аутентификация хранилища пользователя LDAP

1. Запустите getкоманду на testLDAPConnectionконечной точке.
2. Укажите параметры запроса bindCredential, bindDn, connectionUrl, и useTruststoreSpi.
3. Установите actionпараметр запроса на testAuthentication.

   Например:

   $ kcadm.sh create testLDAPConnection -s action=testAuthentication -s bindCredential=secret -s bindDn=uid=admin,ou=system -s connectionUrl=ldap://localhost:10389 -s useTruststoreSpi=always

Добавление картографов

Добавление жестко запрограммированной роли LDAP-картографа

1. Запустите createкоманду на componentsконечной точке.
2. Установите providerTypeатрибут на org.Tuxedo SSO.storage.ldap.mappers.LDAPStorageMapper.
3. Установите parentIdатрибут на идентификатор экземпляра провайдера LDAP.
4. Установите providerIdатрибут на hardcoded-ldap-role-mapper. Убедитесь, что вы указали значение roleпараметра конфигурации.

   Например:

   $ kcadm.sh создать компоненты -r demorealm -s имя=hardcoded-ldap-role-mapper -s providerId=hardcoded-ldap-role-mapper -s providerType=org.Tuxedo SSO.storage.ldap.mappers.LDAPStorageMapper -s parentId=b7c63d02-b62a-4fc1-977c-947d6a09e1ea -s 'config.role=["realm-management.create-client"]'

Добавление картографа MS Active Directory

1. Запустите createкоманду на componentsконечной точке.
2. Установите providerTypeатрибут на org.Tuxedo SSO.storage.ldap.mappers.LDAPStorageMapper.
3. Установите parentIdатрибут на идентификатор экземпляра провайдера LDAP.

4. Установите providerIdатрибут на msad-user-account-control-mapper.

   Например:

   $ kcadm.sh создать компоненты -r demorealm -s имя=msad-user-account-control-mapper -s providerId=msad-user-account-control-mapper -s providerType=org.Tuxedo SSO.storage.ldap.mappers.LDAPStorageMapper -s parentId=b7c63d02-b62a-4fc1-977c-947d6a09e1ea

Добавление атрибута пользователя LDAP mapper

1. Запустите createкоманду на componentsконечной точке.
2. Установите providerTypeатрибут на org.Tuxedo SSO.storage.ldap.mappers.LDAPStorageMapper.
3. Установите parentIdатрибут на идентификатор экземпляра провайдера LDAP.
4. Установите providerIdатрибут на user-attribute-ldap-mapper.

   Например:

   $ kcadm.sh создать компоненты -r demorealm -s имя=атрибут-пользователя-ldap-mapper -s providerId=атрибут-пользователя-ldap-mapper -s providerType=org.Tuxedo SSO.storage.ldap.mappers.LDAPStorageMapper -s parentId=b7c63d02-b62a-4fc1-977c-947d6a09e1ea -s 'config."атрибут.пользователя.модели"=["email"]' -s 'config."атрибут.ldap"=["mail"]' -s 'config."только.чтение"=["false"]' -s 'config."всегда.считывать.значение.из.ldap"=["false"]' -s 'config."обязателен.в.ldap"=["false"]'

Добавление группового LDAP-картографа

1. Запустите createкоманду на componentsконечной точке.
2. Установите providerTypeатрибут на org.Tuxedo SSO.storage.ldap.mappers.LDAPStorageMapper.
3. Установите parentIdатрибут на идентификатор экземпляра провайдера LDAP.
4. Установите providerIdатрибут на group-ldap-mapper.

   Например:

```
$ kcadm.sh создать компоненты -r demorealm -s имя=group-ldap-mapper -s
providerId=group-ldap-mapper -s providerType=org.Tuxedo
SSO.storage.ldap.mappers.LDAPStorageMapper -s parentId=b7c63d02-b62a-
4fc1-977c-947d6a09e1ea -s 'config."groups.dn"=[]' -s
'config."group.name.ldap.attribute"=["cn"]' -s
'config."group.object.classes"=["groupOfNames"]' -s
'config."preserve.group.inheritance"=["true"]' -s
'config."membership.ldap.attribute"=["member"]' -s
'config."membership.attribute.type"=["DN"]' -s 'config."groups.ldap.filter"=[]' -s
'config.mode=["LDAP_ONLY"]' -s
'config."user.roles.retrieve.strategy"=["LOAD_GROUPS_BY_MEMBER_ATTRI
BUTE"]' -s 'config."mapped.group.attributes"=["admins-group"]' -s
'config."drop.non.existing.groups.during.sync"=["false"]' -s
'config.roles=["admins"]' -s 'config.groups=["admins-group"]' -s 'config.group=[]'
-s 'config.preserve=["true"]' -s 'config.membership=["member"]'
```

Добавление полного имени LDAP-картографа

1. Запустите createкоманду на componentsконечной точке.
2. Установите providerTypeатрибут на org.Tuxedo SSO.storage.ldap.mappers.LDAPStorageMapper.
3. Установите parentIdатрибут на идентификатор экземпляра провайдера LDAP.
4. Установите providerIdатрибут на full-name-ldap-mapper.

   Например:

   ```
   $ kcadm.sh создать компоненты -r demorealm -s имя=полное-имя-ldap-mapper
   -s providerId=полное-имя-ldap-mapper -s providerType=org.Tuxedo
   SSO.storage.ldap.mappers.LDAPStorageMapper -s parentId=b7c63d02-b62a-
   4fc1-977c-947d6a09e1ea -s 'config."ldap.full.name.attribute"=["cn"]' -s
   'config."read.only"=["false"]' -s 'config."write.only"=["true"]'
   ```

Операции аутентификации

Установка политики паролей

1. Установите атрибут области passwordPolicyуна выражение перечисления, которое включает в себя конкретный идентификатор поставщика политики и необязательную конфигурацию.
2. Используйте следующий пример для установки политики паролей на значения по умолчанию. Значения по умолчанию включают:
    - 210 000 итераций хеширования
    - по крайней мере один специальный символ
    - по крайней мере один заглавный символ
    - по крайней мере одна цифра
    - не быть равным пользователюusername
    - быть длиной не менее восьми символов

    $ kcadm.sh update realms/demorealm -s 'passwordPolicy="hashIterations и specialChars и upperCase и digits и notUsername и length"'
3. Чтобы использовать значения, отличные от значений по умолчанию, передайте конфигурацию в скобках.
4. Используйте следующий пример для установки политики паролей:
    - 300 000 итераций хэша
    - не менее двух специальных символов
    - не менее двух заглавных букв
    - не менее двух строчных букв
    - не менее двух цифр
    - быть длиной не менее девяти символов
    - не быть равным пользователюusername
    - не повторять по крайней мере четыре изменения назад

```
$ kcadm.sh update realms/demorealm -s
'passwordPolicy="hashIterations(300000) и specialChars(2) и upperCase(2)
и lowerCase(2) и digits(2) и length(9) и notUsername и
passwordHistory(4)"'
```

Получение текущей политики паролей

Текущую конфигурацию области можно получить, отфильтровав все выходные данные, за исключением passwordPolicyатрибута.

Например, отображение passwordPolicyдля demorealm.

$ kcadm.sh получить области/демореальные области --fields паральПолитика

Список потоков аутентификации

Запустите getкоманду на authentication/flowsконечной точке.

Например:

$ kcadm.sh получить аутентификацию/потоки -r demorealm

Получение определенного потока аутентификации

Запустите getкоманду на authentication/flows/FLOW_IDконечной точке.

Например:

$ kcadm.sh получить аутентификацию/потоки/febfd772-e1a1-42fb-b8ae-00c0566fafb8 -r demorealm

Список выполнений для потока

Запустите getкоманду на authentication/flows/FLOW_ALIAS/executionsконечной точке.

Например:

$ kcadm.sh получить аутентификацию/потоки/Копировать%20из%20браузера/выполнения -r demorealm

Добавление конфигурации к исполнению

1. Получите исполнение для потока.
2. Обратите внимание на идентификатор потока.
3. Запустите createкоманду на authentication/executions/{executionId}/configконечной точке.

Например:

$ kcadm.sh create "authentication/executions/a3147129-c402-4760-86d9-3f2345e401c7/config" -r demorealm -b '{"config":{"x509-cert-auth.mapping-source-selection":"Сопоставить SubjectDN с регулярным выражением","x509-cert-auth.regular-expression":"(.*?)(?:$)","x509-cert-auth.mapper-selection":"Пользовательский атрибут Mapper","x509-cert-auth.mapper-selection.user-attribute-name":"usercertificate","x509-cert-auth.crl-checking-enabled":"","x509-cert-auth.crldp-checking-enabled":false,"x509-cert-auth.crl-relative-path":"crl.pem","x509-cert-auth.ocsp-checking-enabled":"","x509-cert-auth.ocsp-responder-uri":"","x509-cert-auth.keyusage":"","x509-cert-auth.extendedkeyusage":"","x509-cert-auth.confirmation-page-disallowed":""},"alias":"my_otp_config"}'

Получение конфигурации для выполнения

1. Получите исполнение для потока.
2. Обратите внимание на его authenticationConfigатрибут, содержащий идентификатор конфигурации.
3. Запустите getкоманду на authentication/config/IDконечной точке.

Например:

$ kcadm получить "authentication/config/dd91611a-d25c-421a-87e2-227c18421833" -r demorealm

Обновление конфигурации для выполнения

1. Получите исполнение для потока.
2. Получить атрибут потока authenticationConfig.

3. Обратите внимание на идентификатор конфигурации из атрибута.
4. Запустите updateкоманду на authentication/config/IDконечной точке.

Например:

$ kcadm update "authentication/config/dd91611a-d25c-421a-87e2-227c18421833" -r demorealm -b '{"id":"dd91611a-d25c-421a-87e2-227c18421833","alias":"my_otp_config","config":{"x509-cert-auth.extendedkeyusage":"","x509-cert-auth.mapper-selection.user-attribute-name":"usercertificate","x509-cert-auth.ocsp-responder- uri":"","x509-cert-auth.regular-expression":"(.*?)(?:$)","x509-cert-auth.crl-checking-enabled":"true","x509-cert-auth.confirmation-page-disallowed":"","x509-cert-auth.keyusage":"","x509-cert-auth.mapper-selection":"Пользовательский сопоставитель атрибутов","x509-cert-auth.crl-relative-path":"crl.pem","x509-cert-auth.crldp-checking-enabled":"false","x509-cert-auth.mapping-source-selection":"Сопоставить SubjectDN с помощью регулярного выражения","x509-cert-auth.ocsp-checking-enabled":""}}'

Удаление конфигурации для выполнения

1. Получите исполнение для потока.
2. Получите атрибут потоков authenticationConfig.
3. Обратите внимание на идентификатор конфигурации из атрибута.
4. Запустите deleteкоманду на authentication/config/IDконечной точке.

Например:

$ kcadm удалить "authentication/config/dd91611a-d25c-421a-87e2-227c18421833" -r demorealm