Инженерный вестник Дона, №3 (2020)
ivdon.ru/ru/magazine/archive/n3y2020/6353

Технология Single Sign On: инструменты централизованной аутентификации для функциональной системы сервисов

А.Ю. Демидова, А.В. Жуков

Петрозаводский государственный университет

Аннотация: В данной статье рассматривается проблема сетевой аутентификации клиента в том случае, когда функциональная web-система состоит из различных сервисов, каждый из которых представлен отдельным сайтом. Особое внимание уделено технологии Single Sign On – единой точке входа как способу достижения централизованной аутентификации. Рассматриваются децентрализованные протоколы аутентификации и предлагается сравнение инструментов, обеспечивающих единых вход. В статье рассматриваются сервера единой точки входа и уточняются их основные возможности.
Ключевые слова: функциональная система сервисов, аутентификация, протокол аутентификации, единая точка входа, сервер единой точки входа, сравнительный анализ.

В сети Интернет существует множество web-сервисов, в том числе, работающих в рамках модели Software as a Service (SaaS). Зачастую они объединяются в комплексы и потребителю преподносятся, как интегрированный инструмент, объединяющий возможности различных систем автоматизации бизнес-процессов. Этот подход позволяет снизить затраты на разработку и сопровождение сервисов, увеличивает жизненный цикл программных комплексов.

К сожалению, при активном использовании нескольких SaaS сервисов, мы сталкиваемся с проблемой дублирования процедуры аутентификации. Это происходит, когда функциональная система состоит из различных web-сервисов, не связанных друг с другом архитектурой, организацией данных. Каждый такой сервис представлен отдельным сайтом и выполняет свою уникальную функцию в рамках системы.

Поскольку сервисы ничем не связаны, мы вынуждены проходить процедуру аутентификации на их сайтах по отдельности. При этом клиент сталкивается с необходимостью помнить логин и пароля от каждого сайта.

Авторами была организована система, состоящая из нескольких сервисов, для которой решалась задача обеспечения единого входа:

- Elgg – выполняет функционал социальной сети,
- Moodle – выполняет функционал предоставления обучающих курсов,
- Rocket.Chat - выполняет функционал предоставления чатов,
- BigBlueButton – выполняет функционал предоставления онлайн-конференций,
- Wordpress – выполняет функционал предоставления ведения блогов,
- NextCloud – выполняет функционал облачного хранилища.

Задачей авторов является поиск решения, направленного на обеспечение единого входа и централизации аутентификации, а также хранения учётных записей пользователей. Нужно упростить сам процесс аутентификации для клиентов и использовать его при организации гетерогенных систем.

Описание технологии Single Sign On

Для обеспечения входа в различные части функциональной системы без повторной аутентификации применяется технология Single Sign On (далее - SSO), предоставляющая возможность аутентификации в поставщике услуг (Service Provider) с помощью доверенной третьей стороны, именуемой поставщиком удостоверений (Identity Provider, Token Provider) [1].

Схема работы данной технологии представлена на рис.1. При использовании поставщика удостоверений поставщики услуг делегируют функцию проверки достоверности сведений о пользователе серверу SSO (токен-провайдер) [2]. Провайдер услуг доверяет выдачу необходимых для

доступа токенов токен-провайдеру и принимает эти токены, как исчерпывающее доказательство успешной аутентификации клиента.

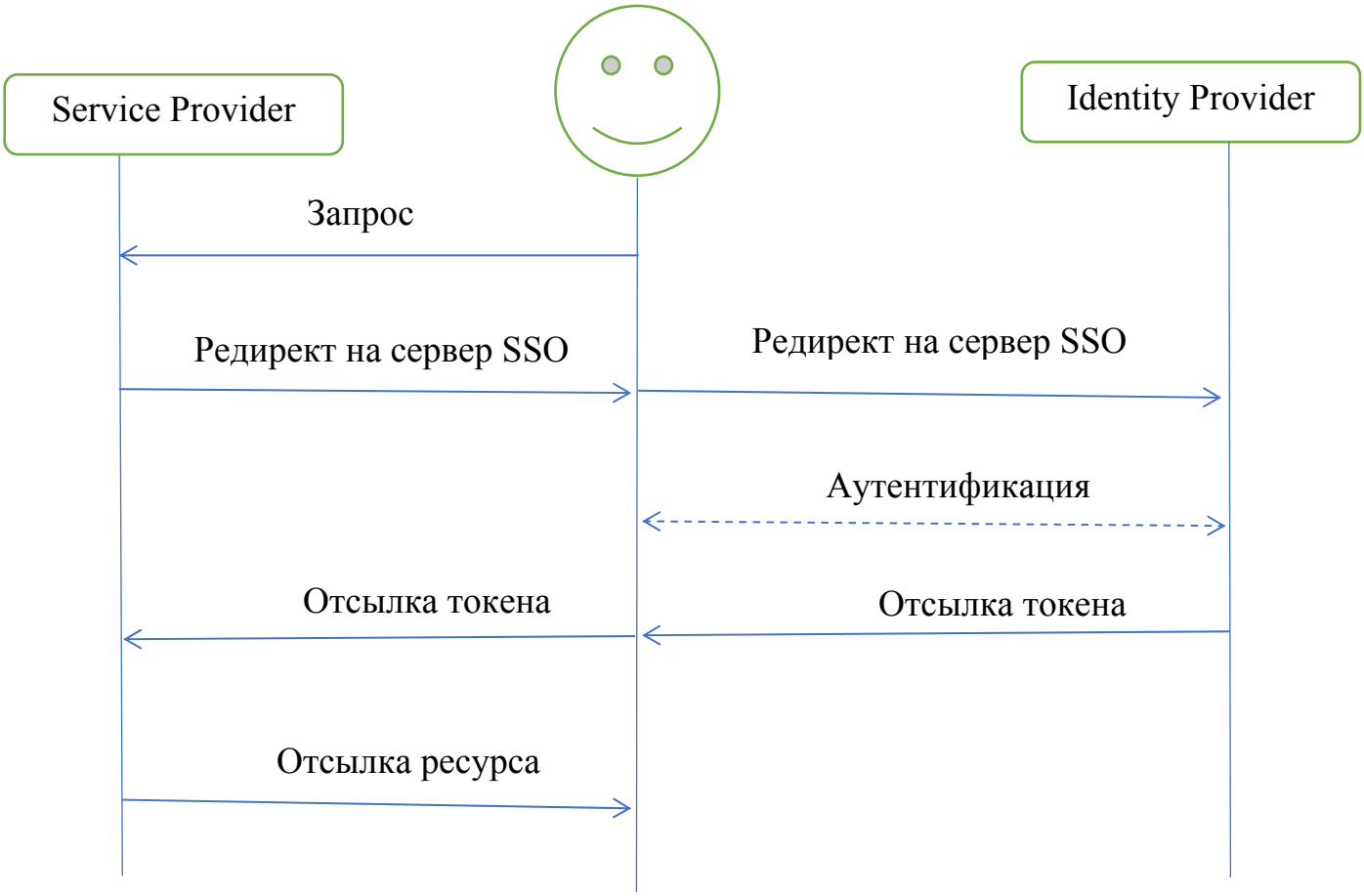

Рис. 1. – Схема работы SSO

При обращении клиента к провайдеру услуг, тот отсылает его к токен-провайдеру за выдачей токена. Токен-провайдер проводит процедуру аутентификации по логину и паролю, проверяет достоверность данных и выдаёт клиенту токен. С этим токеном клиент вновь обращается к провайдеру услуг, получая, таким образом, доступ к сервису. При случае системы из нескольких провайдеров услуг каждый из них принимает единственный токен, позволяя в таком случае клиенту проходить процедуру аутентификации лишь один раз у токен-провайдера [3].

Реализация такой технологии своими силами требует задействовать много ресурсов, поэтому в данной ситуации наилучшим решением будет использование стороннего сервера SSO, который хранит учётные записи пользователей и берёт на себя проведение процедуры аутентификации, а также работает с несколькими общепринятыми протоколами аутентификации. При данном решении необходимо соотнести возможности работы сервера SSO с подходящими нашей задаче протоколами.

Сравнительный анализ сетевых децентрализованных протоколов аутентификации

На сегодняшний момент существует множество протоколов аутентификации в сети Интернет. Каждый из них обладает различными механизмами передачи данных, структурой и сложностью – начиная от базовой и заканчивая многофакторной.

Следует отличать аутентификацию от идентификации и авторизации. Аутентификация – процедура проверки подлинности субъекта, в то время как остальные процедуры – распознавание субъекта по идентификатору и предоставление субъекту определённых прав соответственно.

Процедура проверки подлинности широко распространена как на персональных компьютерах, так и в сети. В нашей работе мы имеем дело с сетевой аутентификацией, поскольку обмен информацией происходит с помощью сети Интернет. При данной процедуре используются протоколы, обеспечивающие защиту линии связи от подмены информации.

Разделяют несколько видов сетевых протоколов аутентификации [4]:

- Базовые – логин и пароль находятся в составе web-запроса;
- Дайджест – пароль передаётся в хешированном виде;
- HTTPS – шифруются все данные, передаваемые между браузером и сервером, а не только логин и пароль;

- Цифровые сертификаты;
- Децентрализованные – процедуру аутентификации берёт на себя внешний (сторонний) сайт;

Для достижения целей данной работы необходимо рассмотреть сетевые протоколы децентрализованной аутентификации. Для сравнительного анализа были отобраны наиболее популярные протоколы:

- SAML
- OpenID
- OAuth2

Данные протоколы необходимо сравнить по нескольким критериям, таким как:

- Формат токена, как оценка удобства работы с данными протокола;
- Наличие авторизации и аутентификации для определения функциональных возможностей протокола;
- Год создания, как определение более совершенных протоколов, в отличие от устаревших;
- Средства передачи сообщений как оценку удобства работы с данными протокола;
- Риски безопасности;
- Ограничения протокола;
- Наилучшее применение протокола;

Результаты сравнительного анализа представлены в таблице № 1 [5].

Таблица № 1

Сравнительный анализ протоколов аутентификации

	OAuth2	OpenId	SAML
	1	2	3
Формат токена	JSON или SAML2	JSON	XML
Наличие авторизации	Да	Нет	Да
Наличие аутентификации	Псевдо-аутентифика-ция	Да	Да
Год создания	2005	2006	2001
Средства передачи сообщений	HTTP	HTTP GET и HTTP POST	HTTP Redirect (GET), SAML SOAP, HTTP POST и другие

	1	2	3
Риски безопасности	Фишинг OAuth 2.0 не поддерживает подпись, шифрование, привязку канала или проверку клиента. Вместо этого он полагается на TLS для обеспечения конфиденциальности [8].	Фишинг Поставщики учётных записей имеют логи входов OpenID, что делает взломанную учетную запись еще большей брешью в сохранении конфиденциальности [6].	Возможность обёртки подписи XML, чтобы выдать себя за любого пользователя [7].

	1	2	3
Наилучшее применение	API авторизация	Единый вход для потреби-тельских приложений	Единый вход для предприятий

Протокол LDAP имеет формат токена, нуждающийся в дополнительном декодировании. Также он в основном применяется для доступа к службе каталогов, а не для единой точки входа.

SAML имеет ограничения для мобильных приложений. Его формат токена требует парсинга (XML) и по размеру больше, чем JSON, однако он широко используется для единого входа предприятий, и потому близок к нашим требованиям.

OAuth2 и OpenId схожи по сравниваемым критериям: оба предоставляют удобный для работы формат токена – JSON, оба работают по HTTP GET и HTTP POST. Однако OpenId имеет преимущество над OAuth2: в отличие от последнего, OpenId создавался именно для единого входа для потребительских приложений, в то время как OAuth2 больше используется для API авторизаций.

По результатам сравнительного анализа можно сделать следующие выводы: протоколы LDAP и RADIUS не подходят для нашей задачи. Протоколы OpenId и OAuth2 являются наиболее предпочтительными для

реализации наших задач. SAML также может быть использован, но скорее как протокол, дополняющий OpenId и OAuth2 на сервере единой точки входа.

После анализа протоколов, предпочтительных для достижения целей данной работы, следующей задачей является выбор сервера единой точки входа, который их предоставляет.

Сравнительный анализ серверов единой точки входа

На текущий момент существует множество серверов, обеспечивающих централизованную аутентификацию и работающих по технологии SSO. Данное программное обеспечение отличается по множеству критериев. Необходимо выбрать, какой сервер будет наиболее оптимальным для достижения цели работы.

Для проведения сравнительного анализа определим критерии, по которым будет проводиться анализ:

- Поддержка протоколов аутентификации – желательно, чтобы сервер работал с как можно более большим количеством протоколов аутентификации, обязательно включая те протоколы, которые были отмечены как основные для работы (OpenId, OAuth2, SAML). Разнообразие протоколов гарантирует, что при подключении нового продукта в систему web-сервисов нам не придётся менять сервер SSO и вновь настраивать его;
- Наличие многофакторной аутентификации – наличие данной поддержки у сервера важно для будущих разработок и совершенствований процедуры аутентификации;
- Наличие админ-панели – данный критерий оценивает удобство использования возможностей сервера;
- Промежуточное ПО – что ещё необходимо для работы сервера;

- Открытый исходный код – один из важнейших критериев, требуются только open-source проекты;
- Возможность коммерческой поддержки;
- Поддержка SCIM (Система междоменного управления идентификацией) – наличие данной поддержки у сервера также важно, поскольку в будущей разработке могут появиться web-сервисы, имеющие разделение на поддомены;
- Мобильное приложение – необязательный критерий, оценивающий лишь возможность работать с мобильными приложениями в рамках SSO.

В сравнении будут участвовать наиболее популярные сервера SSO:

- Aerobase - платформа с открытым исходным кодом для IAM, специализирующаяся на SSO, строгой и адаптивной аутентификации, контроле доступа, управлении учетными записями и обеспечении идентификации, безопасности API и микросервисов и регулировании конфиденциальности. Заявляет о коммерческой поддержке в режиме 24/7.
- Keycloak - сервер для SSO и хранения учётных записей. Этот проект сообщества JBoss находится под управлением компании Red Hat, которая использует его в качестве основного проекта. Первый выпуск Keycloak был в сентябре 2014 года. Среди многих функций Keycloak включают в себя регистрацию пользователя, SSO, двухфакторную аутентификацию, интеграцию с LDAP и многое другое. Не имеет коммерческой поддержки.
- WSO2 - поставщик технологий с открытым исходным кодом, основанный в 2006 году. Он предлагает корпоративную

платформу для интеграции интерфейсов прикладного программирования (API), приложений и веб-сервисов локально и через Интернет. Был основан Сандживой Вераварaной и Полом Фримантлом в августе 2005 года при поддержке компании Intel Capital. Продукты WSO2 выпускаются под лицензией Apache License Version 2. Среди предоставляемых функций – управление доступом и учётными записями и смарт-аналитика.

- Gluu – сервер для центральной аутентификации и авторизации для web и мобильных приложений. Одноимённая компания была основана в 2009 году Майком Шварцем. Включает в себя SSO, строгую аутентификацию, управление доступом, управление учётными записями, интеграцию с LDAP, а также обеспечивает широкую коммерческую поддержку. Одна из особенностей – предоставляемый изначально адаптивный дизайн для всех устройств при входе в приложения, использующие Gluu для аутентификации.
- CAS – проект с открытым исходным кодом для ESSO компании Apereo. В основном ориентирован на корпоративные услуги регистрации и аутентификации. Обеспечивает интеграцию с LDAP, двухфакторную аутентификацию, готовую интеграцию с рядом различных проектов (в т.ч. и Moodle) и коммерческую поддержку.

Результаты сравнительного анализа представлены в таблице № 2 [9].

Таблица № 2

Сравнительный анализ серверов SSO

	Aerobase	Keycloak	WSO2 [10]	Gluu	CAS [11]
	1	2	3	4	5
Поддержка протоколов аутентификации	SAML, OpenID Connect, CAS	SAML, OpenID Connect, OAuth2	SAML, OAuth2	SAML, OpenID Connect, OAuth2, CAS	CAS, SAML, OAuth2, OpenID Connect
Многофакторная аутентификация	да	да	да	да	да
	1	2	3	4	5
Наличие админ-панели	да	да	да	да	да
Открытый исходный код	да	да	да	да	да
Коммерческая поддержка	да	нет	да	да	да
Поддержка SCIM (Система междоменного управления идентификацией)	нет	нет	да	да	да

Мобильное приложение	нет	нет	нет	Super Gluu	нет

По данным таблицы 2 можно сделать выбор в пользу сервера Gluu. Большая часть серверов имеет одинаковые значения критериев, но в совокупности у Gluu самая лучшая совокупность данных значений.

Остальные сервера имеют недостатки. К примеру, Aerobase и Keycloak не имеет поддержки SCIM. WSO2 не обладает разнообразием в протоколах аутентификации, которые он поддерживает. И, наконец, у CAS коммерческая поддержка обеспечивается третьей стороной, что не гарантирует адекватной, качественной и постоянной поддержки.

Заключение

Технология SSO является одним из лучших способов решения проблемы аутентификации в системе web-сервисов, не в последнюю очередь потому, что для её применения уже создано много готовых возможностей. После проведения сравнительных анализов этих возможностей, было создано руководство по выбору готовых серверов SSO и протоколов аутентификации, зависящих от потребностей функциональной системы.

Литература

1. Принципы аутентификации по протоколу Kerberos // ITband.ru. URL: itband.ru/2010/12/kerberos1/ (дата обращения: 1 февраля 2020).

2. Neuman C., Yu T., Hartman S., Raeburn K. The Kerberos Network Authentication Service. MIT, 2005. 138 p.

3. Zhu L., Tung B. Public Key Cryptography for Initial Authentication in Kerberos. Microsoft Corporation, 2006. 42 p.

4. Выростков Д. Обзор способов и протоколов аутентификации в веб-приложениях // URL: habr.com/ru/company/dataart/blog/262817/ (дата обращения: 3 февраля 2020).

5. Fitzpatrick B., Hardt D., Recordon D. OpenID Authentication 2.0 Specification // OpenID Foundation URL: openid.net/specs/openid-authentication-2_0.html (дата обращения: 5 февраля 2020).

6. Hardjono T. OASIS Security Services (SAML) // OASIS URL: oasis-open.org/committees/tc_home.php?wg_abbrev=security (дата обращения: 5 февраля 2020).

7. Hardt D. The OAuth 2.0 Authorization Framework. Microsoft, 2012. 76 p.

8. Lightfoot J. Authentication and Authorization: OpenID vs OAuth2 vs SAML // URL: spin.atomicobject.com/2016/05/30/openid-oauth-saml/ (дата обращения: 5 февраля 2020).

9. Comparison Of Free And Open Source Single Sign On Solutions // OpenAppStack URL: openappstack.net/2019/01/31/comparison-of-free-and-open-source-single-sign-on-solutions.html (дата обращения: 13 февраля 2020).

10. Chinnici R., Moreau J. Web Services Description Language (WSDL) // W3C URL: w3.org/TR/2006/CR-wsdl20-20060327/wsdl20-z.pdf (дата обращения: 10 февраля 2020).

11. JASIG CAS Protocol Page // Apereo URL: w3.org/TR/2006/CR-wsdl20-20060327/wsdl20-z.pdf (дата обращения: 5 января 2020).

References

1. Printsipy autentifikatsii po protokolu Kerberos [Kerberos Authentication Principles] ITband.ru URL: itband.ru/2010/12/kerberos1/

2. Neuman C., Yu T., Hartman S., Raeburn K. The Kerberos Network Authentication Service. MIT, 2005. 138 p.

3. Zhu L., Tung B. Public Key Cryptography for Initial Authentication in Kerberos. Microsoft Corporation, 2006. 42 p.

4. Vyrostkov D. Obzor sposobov i protokolov autentifikatsii v veb-prilozheniyakh [Overview of authentication methods and protocols in web applications] URL: habr.com/ru/company/dataart/blog/262817/

5. Fitzpatrick B., Hardt D., Recordon D. OpenID Authentication 2.0 Specification OpenID Foundation URL: openid.net/specs/openid-authentication-2_0.html

6. Hardjono T. OASIS Security Services (SAML) OASIS URL: oasis-open.org/committees/tc_home.php?wg_abbrev=security

7. Hardt D. The OAuth 2.0 Authorization Framework. Microsoft, 2012. 76 p.

8. Lightfoot J. Authentication and Authorization: OpenID vs OAuth2 vs SAML URL: spin.atomicobject.com/2016/05/30/openid-oauth-saml/

9. Comparison Of Free And Open Source Single Sign On Solutions OpenAppStack URL: openappstack.net/2019/01/31/comparison-of-free-and-open-source-single-sign-on-solutions.html

10. Chinnici R., Moreau J. Web Services Description Language (WSDL) W3C URL: w3.org/TR/2006/CR-wsdl20-20060327/wsdl20-z.pdf

11. JASIG CAS Protocol Page Apereo URL: w3.org/TR/2006/CR-wsdl20-20060327/wsdl20-z.pdf