

УДК 004.42

## Использование CAS для проведения аутентификации в корпоративных приложениях на примере WordPress

*Рыбальченко М.А., студент*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

*Научный руководитель: Остриков С.П., к.т.н*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

[ostrikov@bmstu.ru](mailto:ostrikov@bmstu.ru)

### Введение

Для обеспечения единого безопасного доступа к приложениям, увеличения производительности труда сотрудников поддерживающих подразделений и снижения нагрузки на пользователей применяется унифицированное средство управления доступом на основе технологии однократной аутентификации Single Sign On (SSO)[1]. Технология Single Sign On - это эффективное решение по управлению доступом пользователей к корпоративным приложениям. За счет однократной прозрачной аутентификации SSO позволяет получить доступ к клиент-серверным, Windows, Java и Web приложениям, упрощает управление учетными записями сотрудников и позволяет соответствовать политикам безопасности. Также возможно использование технологии SSO для проведения аутентификации при подключении к различным сервисам.[2]

Существует множество реализаций SSO, одной из которых является Jasig CAS. К преимуществам использования данного решения следует отнести открытый исходный код продукта (*OpenSource*), поддержка нескольких реализаций LDAP, простота настройки и активное сообщество. Также несомненным преимуществом является наличие клиентов под многие популярные платформы, такие как:

- **Drupal, Joomla и WordPress** – системы управления содержимым (CMS);
- **TikiWiki, MediaWiki** – движки для веб-сайтов, работающих по технологии “wiki”;
- **Redmine, Bugzilla, Mantis** – системы для учета и контроля ошибок ПО.

Также имеется множество библиотек для различных платформ и языков программирования, таких как PHP, .NET, Java, Python, Scala, Perl, Ruby On Rails, Erlang и другие, что позволяет разработчикам быстро внедрить аутентификацию с использованием CAS сервера в свои приложения. Поскольку протокол CAS является открытым и хорошо документирован, то возможно написание собственных библиотек работы с CAS сервером для любых других платформ и языков программирования.

### Процесс выполнения аутентификации используя CAS сервер

Существует несколько версий протокола CAS, в данном разделе рассмотрен процесс выполнения аутентификации, работающий по протоколу второй версии CAS v2. Подробно об этом протоколе можно прочитать на портале Jasig CAS: <http://www.jasig.org/cas/protocol>. Ниже приведена схема процесса удачной авторизации в CAS:

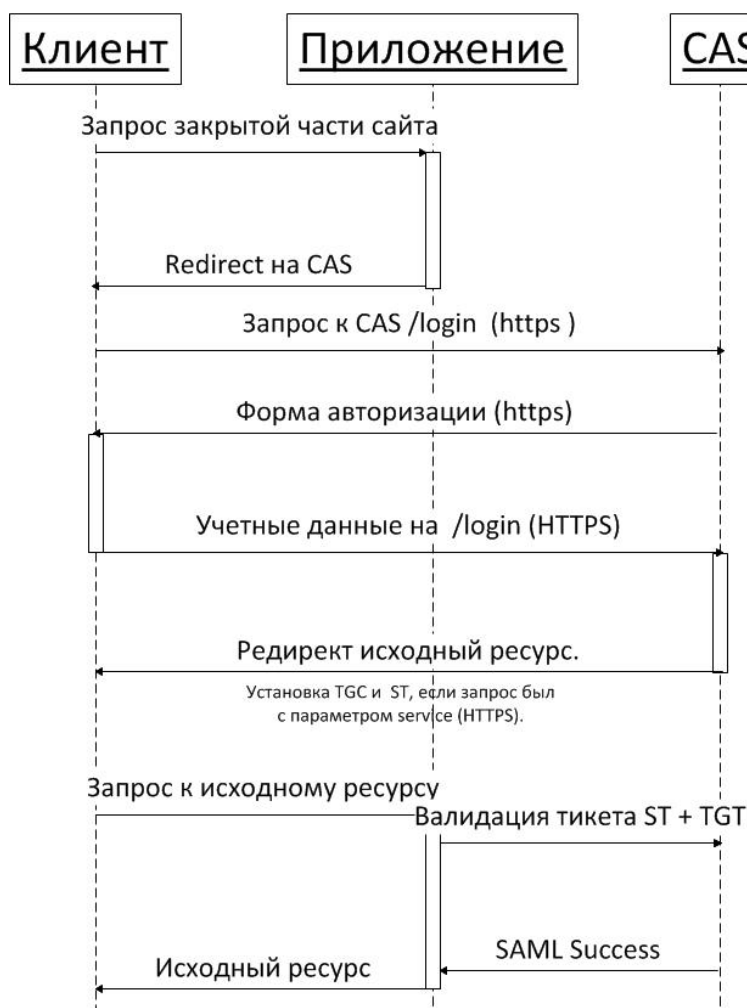


Рис. 1. Диаграмма взаимодействия клиента и CAS сервера при успешном выполнении авторизации

Обозначения на схеме:

- **Клиент** — браузер или программа пользователя;
- **Приложение** — CAS клиент, использующий для авторизации CAS сервер;
- **CAS** — сервер или кластер с развернутым на нем SSO сервисом JasigCAS;
- **ST — Service Ticket**. Представляет собой строку, которая используется как учетные данные клиента для доступа к сервису. Клиент получает его в ответ на предоставленные CAS серверу, учетные данные и идентификатор сервиса;
- **TGT (Ticket Granting Ticket) и TGC (Ticket Granting Cookie)**. TGT представляет собой строку, служащую индикатором авторизованного состояния клиента. После авторизации она заменяет собой учетные данные клиента. Значение TGT хранится в TGC установленной сервером CAS, после успешной авторизации клиента;
- **LT (Login Ticket)** - строка, передающаяся вместе с учетными данными клиента. Используется для того, чтобы исключить повторную обработку учетных данных;
- **SAML (The Security Assertion Markup Language)** — основанный на XML язык, разработанный OASIS.

Следует обратить внимание, что все запросы к SSO серверу должны быть отправлены по протоколу HTTPS. Это не требование протокола CAS, однако, без этого невозможно обеспечить безопасность учетных данных пользователя.

Еще одной важной особенностью протокола CAS является то, что во время проверки значений ST + TGT, CAS клиент, установленный в авторизуемом сервисе, останавливает исходный запрос и создает новый. Т.е. в этот момент клиентом для CAS сервера является не браузер пользователя, а авторизуемый сервис. Это значит, что для создания HTTPS соединения сервис должен быть правильно настроен. Обработка исходного запроса возобновиться только после завершения проверочного[3].

### **Использование CAS аутентификации в WordPress**

В качестве примера внедрения CAS аутентификации в корпоративном приложении был использован сайт, основанный на CMS WordPress. Модульная система расширения функционала, используемая в WordPress, очень удобна и позволяет быстро добавлять новые возможности на сайт. В корпоративной среде WordPress встречается в качестве

основы для внутреннего или внешнего портала организации, систем workflow, helpdesk и прочих.

Также, как было сказано ранее, для WordPress существуют готовые решения реализации CAS клиентов, которые легко внедряются в любой WordPress проект, предоставляя возможность быстрого добавления SSO аутентификации.

Среди множества CAS клиентов для WordPress был выбран CAS Maestro: <https://wordpress.org/plugins/cas-maestro/>. Отличительной особенностью этого модуля являются широкие возможности настройки регистрации пользователей, в том числе с указанием ролей пользователей, а также настройки интеграции в LDAP сервером для получения дополнительных пользовательских данных.

### Требования к окружению

Модуль CAS Maestro использует библиотеку phpCAS для установления подключения к CAS серверу, получения и проверки ticket пользователя. Для работы этой библиотеки требуется наличие в системе следующих установленных приложений и модулей:

- **Веб-сервер** – Apache, IIS или любой другой с поддержкой PHP;
- **PHP** версии 5.0 и выше, скомпилированный со следующими ключами:
  - **--with-curl** – поддержка CURL, требуется для доступа к прокси-серверам.
  - **--with-openssl** - поддержка SSL, требуется для проверки ticket при установке HTTPS соединения;
  - **--with-dom** – поддержка DOM, требуется для обработки XML ответа от CAS сервера;
  - **--with-zlib** – поддержка Zlib, требуется для DOM библиотеки.
- **CURL** библиотека версии 7.5 и выше, скомпилированная с поддержкой SSL.

Данные модули и библиотеки распространяются под большинство современных операционных систем и могут быть установлены в Linux, Windows и MacOS.

### Добавление нового сервиса в панели Jasig CAS

Чтобы сайт имел возможность проводить аутентификацию на CAS сервере, необходимо указать данные сайта в панели управления Jasig CAS. Для этого требуется зайти в панель управления Jasig CAS по адресу <http://<cas-host>:<port>/cas/services>, ввести

данные учетной записи администратора, установленные при настройке Jasig CAS, и выбрать вкладку «Add new service». Содержимое этой вкладки представлено на рисунке:

JA-SIG Central Authentication Service

## Services Management

Add New Service | Manage Services

### Add New Service

**ADD NEW SERVICE**

Please make sure to commit your changes by clicking on the Save Changes button at the bottom of the page

Name:

Service Url:  You can use Ant-style Pattern Matching

Description:

Theme Name:

Status: ☒ Enabled ☒ Allowed to proxy ☒ SSO Participant ☐ Anonymous Access

Attributes:  uid eduPersonAffiliation groupMembership

[Save Changes](#) or [Cancel](#)

Links to CAS Resources: [Home Page](#) [Wiki](#) [Issue Tracker](#) [Mailing Lists](#)

Copyright © 2007 JA-SIG. All rights reserved.

Done

Рис. 2. Вкладка добавления нового сервиса в панели управления CAS сервером

При добавлении нового сервиса в панели управления Jasig CAS необходимо заполнить следующие поля:

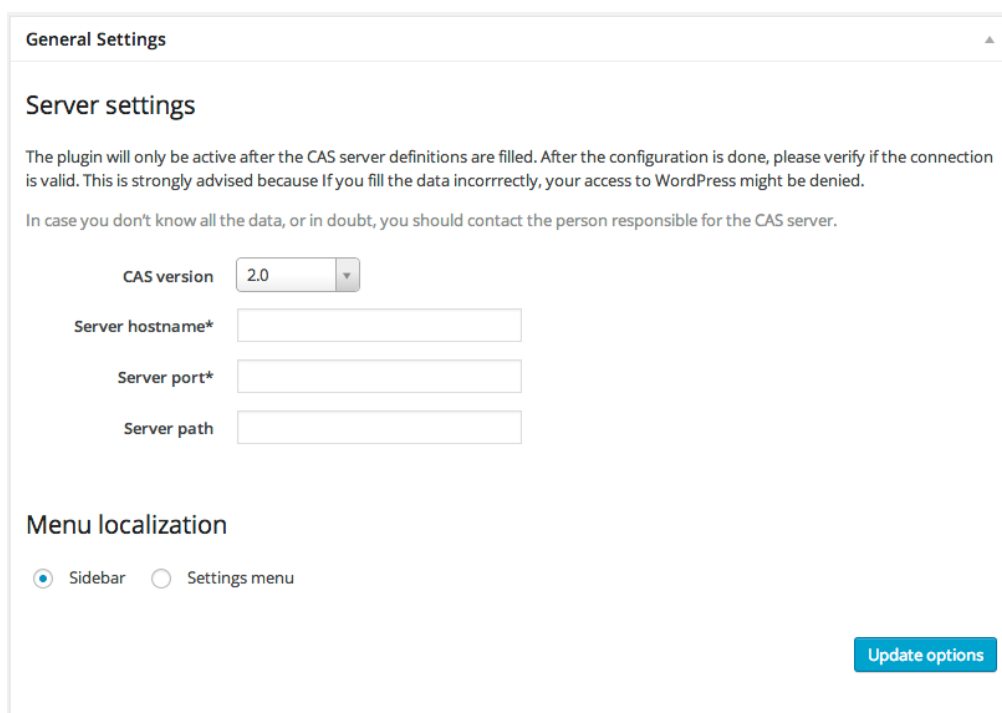
- **Name** – имя добавляемого сервиса. Например, «*My WordPress site*»
- **Service URL** – адрес сервиса. Например, «*http://my-word-press.com/site/\*\**». В данной строке возможно использование шаблонов Ant (<http://ant.apache.org/manual/dirtasks.html#patterns>). Таким образом две звездочки «*\*\**» в конце адреса показывают, что аутентифицированные пользователи могут получать доступ ко всем внутренним каталогам сайта «*site*». Для доступа пользователей как по HTTP так и по HTTPS в строке адреса можно указать как «*http\*://my-word-press.com/site/\*\**»;
- **Description** – описание сервиса. Например, «*Ivanov A. personal page*».

После создания нового сервиса в панели управления Jasig CAS, этот сервис сможет проводить аутентификацию с использованием текущего CAS сервера.

## Установка и настройка плагина CAS Maestro

Установить плагин CAS Maestro можно из общего репозитория WordPress с использованием админ панели. Для этого достаточно зайти в админ панель WordPress по адресу <http://<your-wordpress-host>/wp-admin>, войти под учетной записью администратора, перейти по вкладке «Плагины» и указать в строке поиска: CAS Maestro. После необходимо в списке результатов поиска напротив требуемого плагина нажать кнопку «Установить». В результате плагин CAS Maestro будет установлен и автоматически активирован в WordPress.

Чтобы осуществлять аутентификацию пользователей с использованием этого плагина, необходимо произвести его настройку, указав параметры CAS сервера, через который будет осуществляться аутентификация.



The screenshot shows the 'General Settings' tab for the CAS Maestro plugin. Under the 'Server settings' section, there is a note: 'The plugin will only be active after the CAS server definitions are filled. After the configuration is done, please verify if the connection is valid. This is strongly advised because If you fill the data incorrectly, your access to WordPress might be denied. In case you don't know all the data, or in doubt, you should contact the person responsible for the CAS server.' Below this, there are four input fields: 'CAS version' (a dropdown menu set to '2.0'), 'Server hostname\*' (a text box), 'Server port\*' (a text box), and 'Server path' (a text box). Under the 'Menu localization' section, there are two radio buttons: 'Sidebar' (which is selected) and 'Settings menu'. At the bottom right, there is a blue button labeled 'Update options'.

Рис. 3. Вкладка настроек плагина CAS Maestro

На вкладке настроек плагина CAS Maestro необходимо заполнить следующие поля:

- **CAS version** – версия протокола CAS, используемая сервером;
- **Server hostname** – адрес CAS сервера в формате <http://<домен-или-IP>>;
- **Server port** – порт CAS сервера. В случае веб-сервера Apache или IIS, по умолчанию 80 или 8080. Для Tomcat порт по умолчанию 8443;
- **Server path** – путь до CAS сервера. В случае адреса CAS сервера <http://localhost:8443/cas/>, необходимо указать “/cas/”.

Указав описанные выше настройки подключения к CAS серверу, можно проверить работоспособность аутентификации. Для этого необходимо выйти из текущего пользователя, и заново выполнить вход на сайт. В случае правильной настройки плагина CAS Maestro пользователь будет перенаправлен на страницу входа CAS сервера, где указав свои логин и пароль, автоматически вернется обратно на сайт, осуществив вход.

### **Выводы**

Технология Single Sign On - это эффективное решение по управлению доступом пользователей к корпоративным приложениям. За счет однократной прозрачной аутентификации SSO позволяет получить доступ к клиент-серверным, Windows, Java и Web приложениям, упрощает управление учетными записями сотрудников и позволяет соответствовать политикам безопасности.

Решение Jasig CAS позволяет быстро интегрировать SSO аутентификацию в существующие приложения за счет наличия большого числа клиентов под различные платформы. Также для взаимодействия с CAS сервером имеется множество созданных сообществом библиотек для различных языков программирования, что позволяет разработчикам создавать клиенты для собственных приложений.

Описанный пример внедрения CAS аутентификации в сайт на основе WordPress показывает простоту интеграции, при минимальном требуемом числе настроек, существующего приложения с Jasig CAS.

### **Список литературы**

1. Компания «Совит». Аутентификация пользователей на основе технологии SSO. Режим доступа: [http://www.sovit.net/articles/technologies/single\\_sign\\_on1/](http://www.sovit.net/articles/technologies/single_sign_on1/) (дата обращения 20.11.2014).
2. Козлов М.В. Модель информационной среды университета на основе единого репозитория сервисов // Молодежный научно-технический вестник. МГТУ им. Н.Э. Баумана. Электрон. Журн. 2014. №01. Режим доступа: <http://sntbul.bmstu.ru/doc/681173.html> (дата обращения 20.11.2014).
3. Попов С.А. Единая авторизация (SSO) средствами JASIG CAS. В двух частях. Часть 1. Режим доступа: <http://habrahabr.ru/company/tcsbank/blog/142407/> (дата обращения 20.11.2014).