CAS Plugin

Plugin Information
View CAS on the plugin site for more information.

> Older versions of this plugin may not be safe to use. Please review the following warnings before using an older version:
>
> - Server-side request forgery
> - Arbitrary code execution vulnerability

This plugin enables use of CAS (Central Authentication Service) as an authentication source for Jenkins, with single sign-on and single sign-out support.

> This plugin is meant to replace the CAS1 Plugin by providing new features along existing ones, while adding support for more protocols (e.g. CAS 2.0/3.0 and SAML 1.1)

Setup

Basic Setup

1. Install the plugin from **Manage Jenkins** > **Manage Plugins** > **Available** > **CAS Plugin**.
2. Go to **Manage Jenkins** > **Configure Global Security**, check **Enable Security** and select **CAS (Central Authentication Service)** as the Security Realm.
3. Next to **CAS Server URL**, enter the base URL to your CAS server, e.g. `https://cas.example.com/cas`
4. Next to **CAS Protocol**, select the protocol to use to communicate with CAS, e.g. **SAML 1.1** if you are using Apereo CAS Server 3.x or higher, or **CAS 3.0** if you are using Apereo CAS Server 4.x or higher.
5. If there are no warnings, click the **Save** button at the bottom and attempt logging in.

Advanced Setup

Additional configuration options are available under the **Security Realm** section:

- **Force Renewal:** when checked, single sign-on is disabled: even if a CAS session is already open, the user will have to provide credentials again to confirm his identity.
- **Enable REST API:** when checked, the CAS REST API will be used to authenticate Jenkins API requests (in addition to Jenkins API keys).
- **Enable Single Sign-Out:** when checked, single sign-out is enabled: whenever the user logs out of CAS (e.g. when logging out of another CAS-enabled application), the corresponding Jenkins session will be destroyed and the local user logged out as well. Note that for this to work, the CAS server must be able to communicate with Jenkins using the service URL that was passed to it during login.

Several protocols implemented by CAS are available in the **CAS Protocol** dropdown (click the **Advanced...** button to reveal more options):

- **CAS 1.0:** a text-based legacy protocol. Custom extensions may provide support for roles, which can be parsed with a Groovy script specified in **Roles Validation Script**.
- **CAS 2.0:** a XML-based protocol. It supports **Proxy Tickets**, allowing external applications already secured with CAS to authenticate in Jenkins without requiring user input or password. Custom extensions may provide support for attributes.
- **CAS 3.0:** a XML-based protocol. It supports **Proxy Tickets**, allowing external applications already secured with CAS to authenticate in Jenkins without requiring user input or password. It fully supports attributes out-of-the-box, without requiring custom extensions. **This is a recommended protocol for Apereo CAS Server 4.x and higher.**
- **SAML 1.1:** a XML-based protocol. It fully supports attributes out-of-the-box, without requiring custom extensions. **This is a recommended protocol for Apereo CAS Server 3.x and higher.**

Attributes are an easy (and recommended) way to add full name and email address information to an authenticated user, as well as roles /groups membership. CAS 1.0 response parsing with a custom Groovy script is made available as a legacy option for backward compatibility with the CAS1 Plugin.

Usage

Access from external/scripted clients

By default, when using the CAS plugin for authentication, you **cannot use a regular username/password** for remote authentication into Jenkins. This is by design, as the CAS protocol does not allow "direct" authentication and works with secure redirections, which are not compatible with remote calls such as SVN or GitHub hooks.

You have two options:

- Use the user's **API token** as the password; you can find it by going to the **Configuration** page of the **Jenkins user** you intend to use for external access. This API token does not expire and you may regenerate it as you need.
- Enable the **REST API** option in the plugin configuration, to use the CAS REST API to process the real username/password. The CAS REST protocol must be enabled server-side for this option to work.

See the following page for more information: Authenticating scripted clients

Jenkins URL when used behind a reverse proxy

When using Jenkins behind a reverse proxy, depending on configuration the URL users get redirected to after authentication may be wrong.
If this is the case:

1. Go to **Manage Jenkins** > **Configure System**.
2. Under **Jenkins Location**, make sure the **Jenkins URL** is valid and can be reached by users. It will be used by CAS to redirect back to Jenkins after authentication.

Troubleshooting

SSL certificate issues

Please see the SSL Troubleshooting and Reference Guide from the CAS Project.

Issue validating SAML 1.1 tickets

If Jenkins systematically fails to validate SAML 1.1 tickets, make sure to check whether the **system clock** of your Jenkins and CAS servers are **synchronized**.
Indeed, the timestamp at which SAML 1.1 tickets were generated is checked when validating them, with a configurable tolerance (see "Time Tolerance" plugin option).
Out-of-sync clocks may log errors such as "skipping assertion that's not yet valid" in Jenkins.

Failure to authenticate external/scripted clients

By default, using normal username/password is not possible from external/scripted clients when using CAS.
You may use an **API token** instead and/or enable the **REST API** support. See "Usage" section above for more details.

Missing group memberships when logging with external/scripted clients

This issue (JENKINS-20064) is fixed in Jenkins 1.556 and higher, provided that the user logged in through the web interface at least once.
This limitation does not apply when the REST API option is enabled along with the real username/password.

Invalid Jenkins URL after logging in through CAS

If Jenkins is behind a reverse proxy, it may not be able to detect its own URL by itself. In this case, you need to manually configure the Jenkins URL.
See "Usage" section above for a solution.

Changelog

Version 1.4.3 (2019-01-21)

- Fixed login redirect loop caused by changes in Jenkins 2.160 and 2.150.2 LTS (SECURITY-901, see 2019-01-16 security advisory)

Version 1.4.2 (2018-06-04)

- Fixed security issue (SECURITY-809, see 2018-06-04 security advisory)

Version 1.4.1 (2017-10-01)

- Fixed NullPointerException in SessionUrlAuthenticationSuccessHandler, that could occur when coming back from CAS on some servlet containers (JENKINS-46993).
- Fixed NullPointerException in Cas10Protocol, when using an empty Groovy role parsing script (JENKINS-45441).

Version 1.4.0 (2017-05-09)

- Fixed security issues related to Groovy script execution in CAS Protocol 1.0 configuration (SECURITY-488, see 2017-04-10 security advisory).

Version 1.3.0 (2016-10-19)

- Updated CAS client version to 3.4.1 with less dependencies and support for CAS Protocol 3.0.
- Added CAS REST API support to authenticate Jenkins API calls with real username/password (thanks to Sebastian Sdorra).
- Bumped minimum Jenkins version to 1.625.3 (and require Java 7).
- Restored compatibility with Jenkins version 2.19.1 when using SAML 1.1 (missing dependency no longer required).

Version 1.2.0 (2015-09-13)

- Updated spring-security and CAS client versions with improved robustness and compatibility (thanks to Waldemar Biller).
- Improved detection of Jenkins root URL.
- Fixed usage of forceRenewal parameter in the ticket validator.

Version 1.1.2 (2014-06-02)

- Better handling of multi-valued attributes during Jenkins user creation/update (thanks to Maxime Besson).
- Changed 'Try again' link in failed login page to be relative instead of absolute (fixes issue when Jenkins is run from sub-uri).

Version 1.1.1 (2012-11-10)

- Redirect to origin URL after authentication (instead of always showing Jenkins home page).
- Show custom error page with proper "Try again" link in case of login failure (e.g. due to an invalid ticket).
- Removed unused AspectJ JARs, reducing the overall plugin size (thanks to Jozef Kotlar).

Version 1.1.0 (2012-09-07)

- Support for CAS 2.0 Proxy Tickets, allowing external applications already secured with CAS to authenticate in Jenkins without requiring user input or password.

Version 1.0.0 (2012-09-05)

- Initial release of the new **CAS Plugin**
- Multiple protocols support: CAS 1.0, CAS 2.0, SAML 1.1
- Custom CAS 1.0 response parsing support
- CAS 2.0 and SAML 1.1 attributes support
- Single Sign-Out support
- Jenkins API Token support (no conflict)