URL Auth SSO Plugin

Plugin Information
View URL Auth on the plugin site for more information.

Allows users to be logged in to Jenkins automatically when they are logged into another site.

Features

- Seamless login to Jenkins when logged into the main site
    - Single Sign On
    - Empowers admins to easily make their own authentication backend
    - Makes it possible to use the same account across all sites owned by the same domain

Requirements

Jenkins

Jenkins version 1.625.3 or newer is required.

How it works

1. This plugin authenticates users via a shared identifying cookie. This is likely to be a session ID (e.g. PHPSESSID) which is shared between the Target URL's domain and Jenkins' domain.
2. The identifying cookie must be shared between the two sites. This is possible for subdomains by setting a cookie's domain to .domain.com (note the leading dot).
3. When a user requests a Jenkins page, their Cookie header is sent to the configurable Target URL as a GET request, which authenticates the user and sends back a JSON response with the user_name, display_name and public_email with status 200 OK. All JSON keys are configurable.
4. If the server at the Target URL cannot authenticate the user with the sent cookies, it will respond with error code 401 Unauthorized. If you want to see this in action, try my version.
5. The user will be authenticated in Jenkins if possible. Their username, display name and email will be set using the data from the JSON request.
6. If the user cannot be authenticated, they will be able to click 'Login' at the top right as normal to be taken to the specified external `Login URL` which will log the user into the SSO service. When the user returns a fresh check will be made to check if the user has just logged in.

Because authentication takes place via cookie, this plugin is designed for sites where the user is already logged into a trusted, parent site. It would be a security risk to share sensitive cookies with third party sites.

Setup Guide

You can find a ready-made example backend server in the sso folder, written with PHP and using GitHub OAuth to facilitate SSO. There are a few simple steps to get this example working on your own server.

1. Drag the sso folder into the Document Root of your webserver.
2. Open the sso/signin.php file and set your Client ID and Client Secret from GitHub. Also set the User Agent header to match your own website address (and purpose).
3. Set your session cookie, PHPSESSID, to be shared across subdomains of your domain. This can be accomplished by setting session.cookie_domain = ".example.com" in your php.ini. If you're using Apache on Linux, this is likely to be located at /etc/php5/apache2/php.ini.
4. Install url-auth-sso-plugin on your Jenkins server. Go to Configure Global Security and change the Security Realm to URL Auth Plugin. Set the Target URL to the path to your data.php file - for example, http://example.com/sso/data.php. Also set the Login URL to the path to your signin.php file - for example, http://example.com/sso/signin.php.
5. (Optional) Change the signin.php file to meet your own needs. There are no limits to what you can do, so long as you set at least $_SESSION ["user_name"] as I have in the example script.