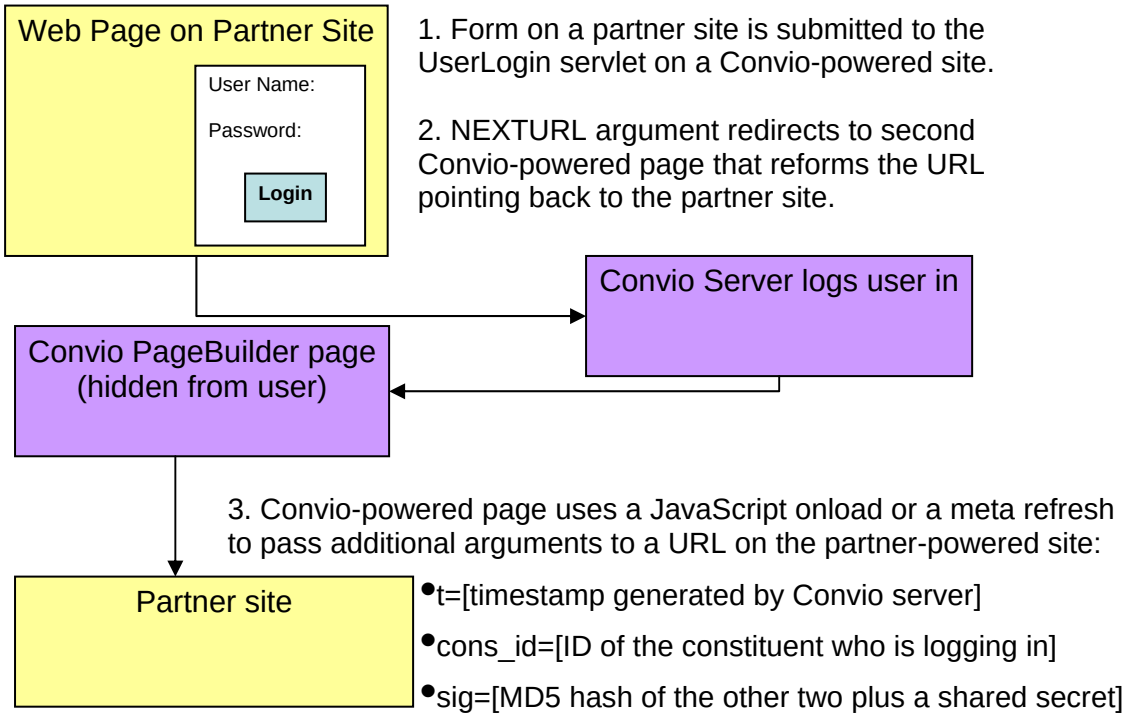
Single Sign-On with Signed URL Redirects

Overview

This document describes a single sign-on protocol with signed URL redirects. A Convio-powered site handles login authentication, and a partner web site hosts the login form. No assumption is made about the two sites sharing cookies.

The following diagram illustrates the basic workflow:



The partner web site hosts a login form that sends its data to the UserLogin servlet on the Convio-powered site. That form includes elements for these parameters:

- USERNAME
- Passwords
- NEXTURL

The user enters the USERNAME and Password. NEXTURL is a hidden element pointing to a Convio-powered PageBuilder page created exclusively for its role in the sign-on process.

If the login credentials are invalid, the user is redirected to the Convio-hosted UserLogin page. That page offers the user several options if he has not registered yet or has forgotten his password. Failed logins cannot be redirected to a page on the partner site -- an aspect of the user experience that individual clients may want tweaked.

The Convio system checks the user's credentials when the form is submitted.

A successful login establishes a user session, and an HTTP redirect invokes the PageBuilder page, which in turn sends a session cookie to the client's browser. The cookie ensures that links from the partner site back to the Convio-powered site share the session. The user must not see the PageBuilder page, which is present only for a JavaScript onload action it contains to redirect the user's browser to the desired landing page on the partner system. The URL for the redirect contains the cons_id (Constituent ID) of the user who has just logged in. The partner system can use appropriate server methods, including calls to the Constituent API, to retrieve or update information about the constituent based on that cons_id.

The return URL is digitally signed to ensure that the cons_id is valid and the login legitimate. The signing process depends on a shared secret exchanged between the partner site and the Convio-powered site. That secret can be stored on Convio as a site data parameter (SDP) or embedded in the PageBuilder page used for the redirection. The signature consists of a timestamp in UTC format and an MD5 hash, which concatenates the cons_id, the timestamp, and the shared secret (in that order).

The partner server has two responsibilities:

1. Validate that the signature is a correct MD5 hash of the other two URL arguments and the shared secret. This ensures the Convio-powered system generated the link.
2. Validate that the timestamp is within a predetermined increment of the system's clock. This can be from 15 seconds, if good clock synchronization is maintained, to 15 minutes if the systems are not synchronized. This validation protects the server against a replay attack.

The partner system can trust the cons_id if both the signature and timestamp are valid.

Code for the Login form

The login form hosted on the partner site follows the model below:

```
<form method="post" action="http://www.foo.org/site/UserLogin">
       <input type="text" name="USERNAME" id="USERNAME" maxlength="60" />
       <input type="password" name="Password" id="Password" maxlength="20" />
       <input type="hidden" name="NEXTURL" id="NEXTURL"
          value="http://www.foo.org/site/PageServer?pagename=sso_redirect" />
</form>
```

The form passes the arguments USERNAME and Password to the Convio-powered UserLogin servlet. The NEXTURL value references the Convio-powered page that will redirect the user back to the partner site.

Code for the PageBuilder page

The PageBuilder page follows the code model below. Note that WYSIWYG editing must be disabled.

```
<!-- Base URL on partner website for redirect -->
[[U0:sso_base_url=http://www.partnersite.com]]

<!-- Shared secret with partner website -->
[[U0:sso_secret=KeepItSafe]]

<!-- ======================================== -->
<!--   No need to change anything below here  -->
<!-- ======================================== -->

<!-- Calculate timestamp and save as sso_timestamp -->
[[U0:sso_timestamp=[[E130:[[S9:timestamp]] 1000 / 1 roundmult]]]]

[[U0:sso_url=[[S80:sso_base_url]][[?[[S80:sso_base_url]]::?::&::?]]cons_id=[[S1
:cons_id]]&t=[[S80:sso_timestamp]]&sig=[[T10:[[S1:cons_id]][[S80:sso_timestamp]
]][[S80:sso_secret]]]]]]

[[?x19x200x::x[[S4]]x::
    <!-- redirect if PageServer or PageNavigator -->
    <script type="text/javascript">
      window.location = "[[S80:sso_url]]"
    </script>
    <noscript>
      Your browser does not support JavaScript, please follow this link to
continue logging in:<br>
      <a href="[[S80:sso_url]]">[[S80:sso_url]]</a>
    </noscript>
::
    <!-- page editor (Don't redirect so we can edit this content) -->
]]
```

This code has a number of Convio “power user” tags. The U0 updates a map stored in the session using its name/value pair. An S80 tag retrieves data from that same session map. Other tags used:

E130 – RPN evaluation of an expression passed to it
S9 – Current time in any of a variety of formats (timestamp is milliseconds since epoch)
S4 – Application ID of the servlet for the current page
T10 – An MD5 hash generated from an expression passed to it

Advanced options

A common use scenario involves returning a user to the original page with its sign-on form in a logged in state. For this to happen, an additional argument has to be passed to

the UserLogin form as a session parameter prefaced, Convio style, with “s_”. To do so, you add a hidden element to the login form:

<input type=”hidden” name=”s_sso_base_url”
value="http://where.you.want.to.return.to" />

The U0 tag with the sso_base_url attribute must be removed from the PageBuilder page.

Logging out from the Convio-powered site

The Convio-powered site should also handle logging out. The best practice is for a logout button to this URL:

http://www.foo.org/site/UserLogin?logout=logout&NEXTURL=partner_logout_url

where partner_logout_url is the URL-encoded value of the location on the partner system where the logout is to be processed. The Convio-powered system logs the user out and then passes control to the partner system to complete the logout and redirect the browser to the desired location.

Maintaining logged-in status on the Convio-powered system

A problem with any single sign-on protocol is maintaining a logged-in state on two systems. Sessions usually expire after a given period of inactivity. On Convio-powered systems, that expiration time is 15 minutes. To keep a Convio-powered session alive, the partner web site needs to include an image tag that references a special servlet on the Convio-powered site:

The image renders a one-pixel square transparent GIF. This should not be embedded in a page until once the user has logged in to the Convio-powered system.

How to have smooth sailing after login

Since the session cookie is set in the user’s browser, any links to the insecure domain (such as, www.foo.org) automatically connect to the correct session. As long as the partner site maintains logged-in status, the flow will appear seamless to the user. Links directly to the secure URL (for example, https://secure2.convio.net/foo) should not be used from the partner site -- a session cookie may not have been set yet. If the partner site links to a page that should be secure, such as a donation form, over an insecure domain, the Convio code will automatically push a redirect to set the secure session cookie.

Access to the Constituent API must be set up in the server code. That code needs to use an API administrator username and password for every call. It must also come from an authorized IP address. Those standard requirements met, the code should be able to use the cons_id from the login redirect response to read and update the constituent’s profile.