Proxied Authentication PRO

You can configure Posit Workbench to participate in an existing web-based single-sign-on authentication scheme using proxied authentication. In this configuration all traffic to Workbench is handled by a proxy server which also handles user authentication. You must configure a firewall or other mechanism to ensure end-user traffic cannot reach Workbench directly.

The fact that Workbench can only receive traffic from the aforementioned proxied server allows it to trust a special HTTP header indicating the username making requests. It will launch and direct traffic to a session owned by that user.

The user must have a local system account on the server. You should set up local system accounts manually and then map authenticating users to these accounts.

Enabling proxied authentication

To enable proxied authentication you need to specify both the `auth-proxy` and `auth-proxy-sign-in-url` settings (the sign-in URL is the absolute URL to the page that users should be redirected to for sign-in). For example:

Listing 1 `/etc/rstudio/rserver.conf`

```
auth-proxy=1
auth-proxy-sign-in-url=http://example.com/sign-in
```

> **!** Important
>
> **Once you enable authentication with a proxy, that becomes the exclusive means of authentication** - you can't concurrently use both PAM and proxied authentication.

> **!** Important
>
> Proxied authentication still requires PAM Sessions and `sssd` to automatically create local system accounts. Without them, local system accounts have to be provisioned manually one-by-one.

Implementing the proxy

Sign in and sign out URLs

The sign in URL should host a page where the user specifies their credentials (this might be for example the main page for an existing web-based authentication system). After collecting and authorizing the credentials the sign in URL should then redirect back to the URL hosting Workbench.

Workbench will redirect to the sign in URL under the following conditions:

- Whenever an HTTP request that lacks the username header is received by the server; and
- When the user clicks the "Sign out" button in the Workbench user interface homepage and there is no Sign Out URL available.

The sign out URL should host a page responsible for finishing the user session in the authentication proxy. If such a URL is available in your proxy, the absolute URL should be configured in Workbench using the setting `auth-proxy-sign-out-url`. When the user clicks the "Sign out" button in the Workbench interface, the browser will be taken to the configured sign out URL.

Listing 2 `/etc/rstudio/rserver.conf`

```
auth-proxy-sign-out-url=http://example.com/sign-out
```

You should be sure in setting up the proxy server that traffic bound for the sign-in and sign-out URLs is excluded from forwarding to Workbench (otherwise it will end up in an infinite redirect loop).

Sign-in delay

During proxied authentication in Workbench, there is a brief transition page that shows the username and some other information. By default this transition happens almost immediately. If you wish to present this page for a longer period of time, you can use the option `auth-proxy-sign-in-delay` to delay the transition for some seconds.

Listing 3 `/etc/rstudio/rserver.conf`

```
auth-proxy-sign-in-delay=4
```

Forwarding the username

When proxying pre-authenticated traffic to Workbench you need to include a special HTTP header (by default `X-RStudio-Username`) with each request indicating which user the request is associated with. For example:

```
X-RStudio-Username: jsmith
```

It's also possible to specify both a system username and a display username (in the case where system accounts are dynamically provisioned and don't convey actual user identity). For example, if the system user is `ruser24` but the displayed username is `jsmith`, you could use:

```
X-RStudio-Username: rsuser24/jsmith
```

> **Note**
>
> It is highly recommended that you *do not use* the default `X-RStudio-Username` header name. The reasons for this are described in the section on Proxy security considerations below.

Rewriting usernames

It may be that the proxy system you are using sends the username in a format that doesn't match that of users on the system, however can be easily transformed to one that does (e.g. it has a standard prefix before the username). If this is the case you can specify the `auth-proxy-user-header-rewrite` option to provide a re-write rule for the inbound header. For example, the following rule strips the prefix "UID-" from a username header:

```
auth-proxy-user-header-rewrite=^UID-([a-z]+)$ $1
```

The format of a re-write rule is a regular expression followed by a space and then a replacement string. The replacement string can reference captured parts of the regular expression using `$1`, `$2`, etc. Consult the Boost Perl Regular Expression Syntax reference for more syntax documentation.

Proxy security considerations

When using proxied authentication, Workbench trusts that the proxy server is the only element in the network capable of sending the special header with the username. Be sure to follow the recommendations below to decrease security risks in your implementation.

Keeping the header name secret

Using the default header name `X-RStudio-Username` may create a security problem if code running behind the proxy (e.g., code within RStudio Pro Sessions) can form requests directly back to the Workbench server. In that case, code running in the session from one trusted user could be used to impersonate another (by simply inserting the header in their request).

To prevent these loopback requests, see the Preventing internal access section below.

It's also recommended to specify a custom header name which is kept secret from end users. This is done by creating a special configuration file (`/etc/rstudio/secure-proxy-user-header`) that contains the name of the header, and then setting it's file permissions so that it's not readable by normal users. For example:

```
sudo sh -c "echo 'X-Secret-User-Header' > /etc/rstudio/secure-proxy-user-header"
sudo chmod 0600 /etc/rstudio/secure-proxy-user-header
```

Preventing remote use of the header

When implementing the proxy it's important to remember that Workbench will always trust the username header to authenticate users. It's therefore critical from the standpoint of security that all requests originating from the proxy have this header set explicitly by the proxy (as opposed to allowing the header to be specified by a remote client). Workbench will reject requests containing multiple conflicting occurrences of the username header.

Preventing internal access

> **Note**
>
> In previous versions, Workbench offered the option `auth-proxy-require-hmac` to require trust signatures from the proxy, though most proxies have no straightforward means for providing this signature. Therefore, this option has been retired. Following the recommendation below mitigates the same security risks previously covered by the option. If your installation used this option, it should removed from `/etc/rstudio/rserver.conf`. Workbench will refuse to start if this option is still present and enabled.

Your Workbench and proxy configuration should be done in a way where it is impossible for anything other than the proxy to make requests to Workbench. Be sure that:

- Workbench is configured to listen on a network interface not accessible internally by other processes by adjusting the option `www-address`.
- The network interface where Workbench is running must have firewall settings to prevent any connection to Workbench other than from the proxy.

> **!** Important
>
> This should not be considered an exhaustive list. Please consult with your security personnel or IT administrators to determine the exact measures to protect Workbench authentication via a proxy.

Troubleshooting with access logs

If you want to see exactly which requests Workbench is receiving and whether they include the expected username information, you can temporarily enable server access logs using the `server-access-log` setting as follows:

Listing 4 `/etc/rstudio/rserver.conf`

```
server-access-log=1
```

After restarting the Workbench server process the following file will contain a record of each HTTP request made to the server along with it's HTTP response code:

```
/var/log/rstudio/rstudio-server/rserver-http-access.log
```

The log file will contain entries that look like this:

```
127.0.0.1 - - [29/Jun/2015:06:30:41 -0400] "GET /s/f01ddf8222bea98a/ HTTP/1.1"
200 91 "http://localhost:8787/s/f01ddf8222bea98a/" "Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.125 Safari/537.36" "jsmith"
```

Note that the very last item in the log file entry is `"jsmith"`. This is the username that Workbench read from the header passed by the proxy server. If this shows up as blank (`"-"`) then your proxy server isn't forwarding the header or using the correct header name in forwarding.

> **Important**
>
> Once you've concluded troubleshooting it's important that you remove the `server-access-log=1` option from the `/etc/rstudio/rserver.conf` file (since this log file is not rotated it will eventually consume a large amount of disk space if you don't remove the option).