SAML Single Sign-On Authentication PRO

Table of contents

Posit Workbench can be configured to authenticate users via SAML. This enables users to log in with their existing Single Sign-On (SSO) credentials and to be automatically authenticated to Workbench whenever they are already logged into their Identity Provider (IdP) account.

Enabling SAML SSO

To enable authentication with SAML you add the `auth-saml` option to the Workbench configuration file `/etc/rstudio/rserver.conf`:

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

```
auth-saml=1
```

❗ Important

Once you enable authentication with SAML that becomes the exclusive means of authentication - you can't concurrently use both PAM and SAML authentication.

❗ Important

SAML 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.

Configuring SAML

The easiest way to setup SAML authentication is to:

1. Configure Workbench with metadata from your IdP
2. Configure the IdP to point to the Workbench's URL
3. Configure additional settings

The identity providers Okta, OneLogin, and Azure have preconfigured entries in their setup UIs for Workbench to make the second step of the configuration easier. For more information on configuring Okta, OneLogin, Azure, please see Configuring SAML in Workbench.

For other IdPs that require more than basic interoperable SAML, see the Advanced options section to be sure your IdP is supported.

If your IdP does not provide its metadata until Workbench is running, see Configuring IdP metadata first.

See these sections for additional setup if they apply:

- Proxy considerations
- SAML with load balancing

Step 1. Configure Workbench with IdP metadata

There are three ways to provide Workbench with the metadata from your IdP:

- Use an Idp Metadata URL online
- Use an Idp Metadata file offline
- Manual configuration of IdP information

For all three approaches, after a successful login the IdP will provide Workbench with an assertion: a securely signed list of attributes about the user.

> ❗ Important
>
> Your IdP must return at least one attribute (or `NameID`) matching the user's account username in the assertion. See PAM basics for details on username handling.

By default, Workbench will look for an attribute called `Username` (case-sensitive). If your IdP uses a different attribute, for example `NameID`, add the option `auth-saml-sp-attribute-username` with the appropriate value.

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

```
auth-saml-sp-attribute-username=NameID
```

> ℹ Note
>
> Workbench expects the IdP metadata to contain the service name (`EntityID`), the Single Sign-On (SSO) URL and the signing certificate.

Option 1. Online - IdP metadata URL

The easiest way to configure Workbench is directly with the IdP metadata URL. This can be done by adding the option `auth-saml-metadata-url`. This option has the advantage of automatically renewing the metadata when it expires but to use it, you'll need to allow Workbench to access the IdP URL directly.

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

```
auth-saml-metadata-url=https://idp.example.com/saml/metadata
```

Option 2. Offline - IdP metadata file

To avoid direct connectivity between your Workbench server and the IdP use an offline setup. First download the metadata from your IdP and upload it to your server. Then, add the option `auth-saml-metadata-path` pointing to the file location within your server. This option requires manual intervention if users can no longer login because the metadata or the signing certificate expired.

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

```
auth-saml-metadata-path=/path/to/saml/metadata.xml
```

> ⚠ Warning
>
> The metadata URL option has precedence over the metadata file path option. You must remove the URL option first before using the file option.

If your IdP requires information about Workbench in order for you to access its metadata, see the [Manual service provider setup] section below for how to obtain this information.

Option 3. Manual identity provider (IdP) setup

If your IdP does not provide metadata or if the metadata does not have all of the information required, use a manual setup. The following required options must be added to your server configuration:

- `auth-saml-idp-entity-id` - A URL to an HTTP(S) endpoint on the IdP, in general the location of its metadata. In very exceptional cases this may not be an URL.
- `auth-saml-idp-sso-url` - A URL to an HTTP(S) endpoint on the IdP to where your server will send authentication requests.
- `auth-saml-idp-sign-cert-path` - The path to a PEM file containing the public trust certificate for verifying the assertions' signatures.

Listing 5 `/etc/rstudio/rserver.conf`

```
auth-saml-idp-entity-id=https://idp.example.com/saml/metadata
auth-saml-idp-sso-url=https://idp.example.com/saml/sso
auth-saml-idp-sign-cert-path=/path/to/saml.cert
```

> ⚠ Warning
>
> The metadata URL and file path options have precedence over the individually configured options. You must remove the metadata options first before using individual settings.

Step 2. Configure your identity provider with Workbench

Once Workbench knows about your IdP, use the administration UI of your IdP to add the URL of your Workbench server and other information it requires.

Preconfigured setups

Workbench has preconfigured entries in Okta, OneLogin and Azure. In some cases, all you need to provide is the URL to your server. Please refer to the documentation on these vendors for more information.

Service provider metadata setup

When your IdP requires a Service Provider (SP) metadata URL, add `/saml/metadata` onto your Workbench server's address like this: `https://server.example.com/saml/metadata`.

If the IdP asks for an IdP metadata file, or requires manual configuration, save the contents of that URL to a file with your browser, curl or wget.

For a manual configuration, look in the text file that is returned for the required values.

> **i** Note
>
> The SAML metadata primarily contains information about the service name (`EntityID`) and the assertion consumer service (ACS) URL.

Configuring IdP metadata first

Some IdPs must be configured with Workbench's metadata before they will allow Workbench to access their metadata. Workbench itself won't start when it's configured to point to an invalid IdP metadata URL. In these situations, you can manually enter the information into the IdP as follows:

- *Workbench Entity ID*: This value is the same URL as the metadata endpoint used for Service provider metadata setup. For example, https://server.example.com/saml/metadata
- *Workbench Assertion Consumer Service URL*: Workbench expects SAML assertions at the `/saml/acs` endpoint. For example, https://server.example.com/saml/acs
- If encryption is used, you will need to provide the encryption certificate used by Workbench, see SAML encryption below. Workbench supports most common forms of encryption used with SAML.
- If your Identity Provider expects signed requests from Workbench, you will need to provide the signing certificate used by Workbench, see SAML request signing below. Any signing algorithm you choose in your IdP must match Workbench's configuration.

- [SAML attributes] as mentioned above.
- Also, information requested about Unsupported SAML options in Workbench should be left blank.

Advanced options

Depending on your IdP capabilities you may need to add a few more options to your server:

- `auth-saml-idp-post-binding`: By default Workbench will redirect to your IdP for authentication requests. With the value `1`, this option makes it use an HTTP POST instead. This option can also be used with a metadata file or URL if your IdP supports both redirect and POST. `auth-saml-sso-initiation`: By default, Workbench will be able to initiate authentication with SAML (SP-initiated) or to accept an ad hoc assertion (IdP-initiated). If you prefer just one of these flows, use this option with either `sp` or `idp` values. When set to `idp`, users will be sent to the configured IdP SSO URL if an SP-initiated flow is attempted.
- `auth-saml-sp-name-id-format`: By default Workbench will accept any NameID Format. Add this options with the values `persistent`, `transient`, `emailaddress`, or `unspecified` to make Workbench request and expect a particular format from the IdP.

> Warning
>
> `auth-saml-sp-name-id-format=transient` and `auth-saml-sp-attribute-username=NameID` will not be accepted as a valid combination. It would lead to undetermined usernames in each attempt.

Here are some examples of valid configurations of the aforementioned advanced options:

Listing 6 `/etc/rstudio/rserver.conf`

```
auth-saml-idp-post-binding=1
auth-saml-idp-sso-url=https://idp.example.com/saml/sso
```

Listing 7 `/etc/rstudio/rserver.conf`

```
auth-saml-sso-initiation=idp
auth-saml-idp-sso-url=https://idp.example.com/login
```

Listing 8 `/etc/rstudio/rserver.conf`

```
auth-saml-sp-name-id-format=persistent
```

SAML encryption

To enable support for encrypted SAML assertions, you will need a key pair in the form of a public certificate file and a private RSA key, both in PEM format.

The following options should be added to your server:

- `auth-saml-sp-encryption-key-path`: The path to a PEM file containing the private RSA key for decrypting the assertion.
- `auth-saml-sp-encryption-cert-path`: The path to a PEM file containing the public certificate for encrypting the assertion. The contents of this file will be present in your server metadata after configured. You may also be asked to upload this certificate to the IdP instead.

Listing 9 `/etc/rstudio/rserver.conf`

```
auth-saml-sp-encryption-key-path=/path/to/saml.key
auth-saml-sp-encryption-cert-path=/path/to/saml.cert
```

> ⚠ Warning
>
> These key pair files are similar to the ones used for SSL/TLS. However, for security reasons you must never use your server's own SSL/TLS key and certificate for SAML encryption.

This example allows the creation of a simple self-signed public certificate and private key pair that can be used for encryption for the server "localhost" (you should use your server public facing hostname instead):

```
openssl req -x509 -newkey rsa:2048 -keyout saml.key -out saml.cert -days 365 -nodes -subj
```

SAML request signing

> ℹ Note
>
> In most situations, SAML authentication request signing is not required or even supported. Be sure your IdP requires signing before using this functionality.

To enable support for signed SAML authentication requests, you need to set a signing method in your server configuration with the option `auth-saml-sp-request-signing-method`. The algorithms `sha1`, `sha256`, or `sha512` are supported. When in doubt, try `sha256` first which offers a good balance between security and compatibility.

Listing 10 `/etc/rstudio/rserver.conf`

```
auth-saml-sp-request-signing-method=sha256
```

> ℹ Note
>
> By using one of the signing methods listed above the Workbench metadata will contain a `"signing"` certificate and will have the attribute `AuthnRequestsSigned` with the `true` value.

You will also need a key pair in the form of a public certificate file and a private RSA key, both in PEM format. If you are using SAML encryption, the already configured encryption key pair will also be used for request signing.

> ❗ Important
>
> Workbench only accepts a single key pair. You can use a key pair for **both** encryption and signing requests or one just for signing. Different key pairs for signing and encryption are not currently supported.

If you are not currently using SAML encryption, the following options should be added to your server:

- `auth-saml-sp-signing-key-path` - The path to a PEM file containing the private RSA key for decrypting the assertion.
- `auth-saml-sp-signing-cert-path` - The path to a PEM file containing the public certificate for encrypting the assertion. The contents of this file will be present in your server metadata after configured. You may also be asked to upload this certificate to the IdP instead.

Listing 11 `/etc/rstudio/rserver.conf`

```
auth-saml-sp-signing-key-path=/path/to/signing.key
auth-saml-sp-signing-cert-path=/path/to/signing.cert
```

> ⚠ Warning
>
> These key pair files are similar to the ones used for SSL/TLS. However, for security reasons you must never use your server's own SSL/TLS key and certificate for SAML encryption.

This example allows the creation of a simple self-signed public certificate and private key pair that can be used for encryption for the server "localhost" (you should use your server public facing hostname instead):

```
openssl req -x509 -newkey rsa:2048 -keyout saml.key -out saml.cert -days 365 -nodes -subj
```

Unsupported SAML options

Workbench supports at least a subset of SAML called Interoperable SAML. Notably, certain functionalities are currently absent:

- Single Logout
- Certificate chain validation
- RelayState URL handling (not part of the SAML standard)

SAML with load balancing

Because Workbench stores SAML authentication context in server memory during the authentication flow, the entire authentication flow must be completed on a single server. If you're using an external load balancer in front of Workbench, you will experience authentication errors if the HTTP requests associated with the authentication flow are not all routed to the same server.

For this reason, you **must enable sticky sessions** in your external load balancer when using SAML authentication. This feature is sometimes called "sticky cookies" or "session affinity". Consult the documentation for your load balancing software for details; for example if you're using the Amazon Web Services Application Load Balancer (AWS ALB), more information can be found in Sticky Sessions for your Application Load Balancer.

Proxy considerations

If you are running Workbench behind a proxy, you will need to configure your proxy in a way that Workbench can tell the SAML IdP to redirect back to the correct location. There are number of options to choose from as described in Running with a Proxy.

The use of the `X-RStudio-Request` header in your proxy is recommended and the only method which works behind a path-rewriting proxy. In this case, the proxy must set the `X-RStudio-Request` header to the exact complete URL as requested by the browser. For example if your proxy was set up to serve Workbench requests at `https://testdomain.com/rstudio/` and an incoming request for `/home` came in, your proxy should set `X-RStudio-Request: https://testdomain.com/rstudio/home` which would allow RStudio to know about the added path prefix `/rstudio`.

If your proxy does not add path prefixes, Workbench is also compatible with two options using commonly available HTTP proxy headers:

- The headers `X-Forwarded-Host`, `X-Forwarded-Proto`, and `X-Forwarded-Port`.
- Or the header `Forwarded` with `host`, and `proto` values.

When using path-rewriting proxies, it's also recommended to use either the header `X-RStudio-Root-Path` or the option `www-root-path` to indicate the path defined for Workbench by the proxy. For example, if your URL to Workbench is `www.example.com/rstudio` your proxy should send the header `X-RStudio-Root-Path: /rstudio` or you should use:

Listing 12 `/etc/rstudio/rserver.conf`

```
www-root-path=/rstudio
```

If none of the headers above are set by the proxy, Workbench will redirect back to the address present in the `Host` header and it will determine the protocol from rserver.conf's `ssl-enabled` setting. Use a value of `1` for `https` and `0` for `http`.

If you are running behind a proxy, it's possible that the proxy terminates the SSL connection. That means that the SAML URL should use `https` but rserver.conf will have `ssl-enabled=0`. In that case, you can directly configure the SAML base URI by setting `auth-sp-saml-base-uri` in `rserver.conf`:

Listing 13 `/etc/rstudio/rserver.conf`

```
auth-saml-sp-base-uri=https://testdomain.com/rstudio/
```

In general, you can use this setting to specify the external base URL to be used in the /saml/metadata for when the www-root-path and ssl-enabled settings, along with the request headers, do not match your proxy server's configuration.

Outgoing proxies

Some SAML authentication features require Workbench to make a call to an external service over HTTP or HTTPS; for example, to perform provider metadata discovery. If your environment requires an HTTP or HTTPS proxy for outbound requests, you must set the appropriate proxy environment variables for Workbench's server process so that it uses the proxy when making the request.

One way to do this is to add the variables to the env-vars file as follows:

Listing 14 /etc/rstudio/env-vars

```
HTTP_PROXY=http://192.168.1.1:8080
HTTPS_PROXY=http://192.168.1.1:8080
NO_PROXY=localhost,192.168.1.10
```

Troubleshooting

Additional information about the SAML flow and the received assertion may be written to the logs. Be sure to configure rserver logs to output debug level messages in /etc/rstudio/logging.conf to see these entries.