PAM Sessions & Kerberos PRO

Although the IDE sessions that users access reside in a browser, they are still backed by real server-side processes and filesystem resources. Because of this, it can make sense to configure Workbench to start a traditional server-side PAM session for each user, much as a remote shell would.

> ℹ Note
>
> Using PAM sessions is distinct from using PAM for authentication to Workbench itself, and works with other authentication methods (OpenID Connect, SAML, and so on).

The most common reason to use PAM sessions is to automatically provision local Linux users from an LDAP or Active Directory provider as they are requested, in conjunction with SSSD (or an alternative like Centrify).

However, organizations sometimes use PAM sessions to manage access to on-premise data sources, such as:

- Mounting remote drives locally, most often Windows SMB/CIFS shares.
- Managing Kerberos tickets, usually to access SQL Server databases

Keep in mind that leaning on these filesystem-based PAM features may have major downsides: in particular, they are generally incompatible with cloud-native features like launching IDE sessions on Kubernetes and modern authentication options like OpenID Connect or SAML.

Finally, it is important to understand that PAM sessions are fundamentally local to a single machine. This means that if you are running in a load-balanced configuration, PAM sessions can and will start on more than one node. The same applies when using the Job Launcher with Kubernetes: PAM sessions may be started on both the server node (for the home page) and inside the containers used to run IDE sessions. With Slurm, PAM sessions may run on both submission and compute nodes.

In all of these cases, consider the PAM profile local to these machines or containers.

PAM session behavior

PAM sessions work slightly differently depending on how Posit Workbench is configured:

- When the Job Launcher is enabled and PAM-based password forwarding is turned off, the user's home page starts a single PAM session on the server node (or one of the nodes, when load balancing is enabled). This is the most common setup, where PAM sessions are used primarily for provisioning user home directories.
- When the Job Launcher is enabled and PAM-based password forwarding is turned on, the user's home page starts a PAM session on a server node and individual IDE sessions start their own PAM sessions – possibly in a container on Kubernetes or on a Slurm compute node. This setup allows for using filesystem-bound PAM mechanisms like remote mounts or Kerberos, but comes with other limitations.
- When the Job Launcher is disabled (and consequently all IDEs except RStudio Pro), each RStudio Pro session will start its own PAM session, as will the user's home page. This configuration was the default in older versions of Workbench that did not support other IDEs, and is consistent with using PAM sessions for user provisioning and for filesystem-bound operations like remote mounts or Kerberos. In addition, when an Rstudio Pro session is forced to suspend by an administrator in this setup, it will also *close* its PAM session. This can be accomplished in one of two ways:
  - By pressing the **Suspend** button on the *Sessions* page of the Administrative Dashboard.
  - By executing a `force-suspend` or `force-suspend-all` command as described in Suspending sessions.

Workbench requires user home directories to be on shared storage when using a load-balanced configuration, Kubernetes, or Slurm – so it is usually only necessary to create them once.

Enabling or disabling PAM sessions

By default PAM sessions are disabled when using the Job Launcher and enabled otherwise. Enable them explicitly by using the `auth-pam-sessions-enabled` setting:

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

```
auth-pam-sessions-enabled=1
```

PAM sessions can sometimes introduce unwanted complexity, including unnecessary environment variables, mounts, or Kerberos tickets, as well as performance problems when starting

IDE sessions. If you don't need these features or the user provisioning provided by SSSD, you may want to disable them:

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

```
auth-pam-sessions-enabled=0
```

Although you may disable PAM sessions completely with this setting, in most cases we recommend using a custom, slimmed-down PAM profile instead. If you are not certain, please contact Posit support.

> ⚠ Warning
>
> When using SSSD to automatically provision local system accounts using LDAP or Active Directory, Workbench relies on PAM sessions configured with `pam_mkhomedir` (or equivalent) to create the home directories of users that have never logged in to the server. Entirely disabling PAM sessions in this scenario may cause permission errors when starting sessions.

PAM session profiles

The behavior of the Workbench server is essentially the same as that of the `su` command – impersonation of a user – so by default it uses the `/etc/pam.d/su` PAM profile. This profile is installed by default on all supported platforms.

Importantly, during startup and shutdown, Workbench must be able to elevate its own permissions to the `root` level to perform various non-user-specific management tasks on the system. Following startup, Workbench then deescalates its permissions back to the user level to handle session operations on behalf of the user. Workbench also cannot retain the passwords used during the authentication phase (with some exceptions, see below). As a result of these requirements, this profile **must** contain a `sufficient` PAM directive that enables authentication as `root` without a user password, via `pam_rootok.so`:

Listing 3 `/etc/pam.d/su`

```
# This allows root to su without passwords
auth       sufficient pam_rootok.so
```

Again, this is the default on all supported platforms.

> **Note**
>
> Some SSSD configurations also require PAM account verification as `root` to present on both the `auth` and `account` directives in the PAM profile (`auth sufficient pam_rootok.so` and `account sufficient pam_rootok.so`). Verify that this is included if you see errors when starting new IDE Sessions.

Using a custom profile

If you need additional PAM session behavior specific to Posit Workbench sessions, you can create a custom PAM profile. For example, if you wanted to use a profile named `rstudio-session` you would add this to Workbench's configuration file:

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

```
auth-pam-sessions-profile=rstudio-session
```

Below is an example of what the custom profile might contain to enable a few common features of PAM sessions (this is based on a modified version of the default `su` profile on Ubuntu):

> **Important**
>
> The order of directives matter. `auth sufficient pam_rootok.so` ends the processing of auth-type modules, so you will want this directive to be placed (1) before other modules that require a password; but (2) after other auth-type modules you want to run regardless.

Using a custom profile with passwords

> **Important**
>
> Because it relies on directly capturing passwords, this feature is only available when using PAM Authentication. If you are using the Job Launcher, you must also configure it to use TLS/SSL, as Workbench will refuse to transmit passwords to IDE sessions over an unencrypted channel.

In the configuration samples above, we rely on `pam_rootok.so` to enable authentication without a password. This is necessary because Workbench doesn't retain the passwords used during the authentication phase – nor is it possible to do so when using, e.g., OpenID Connect or SAML to log in.

Listing 5 `/etc/pam.d/rstudio-session`

```
# This allows root to su without passwords (this is required)
auth       sufficient pam_rootok.so

# This module parses environment configuration file(s)
# and also allows you to use an extended config
# file /etc/security/pam_env.conf.
# parsing /etc/environment needs "readenv=1"
session    required   pam_env.so readenv=1

# Locale variables are also kept into /etc/default/locale in etch
# reading this file *in addition to /etc/environment* does not hurt
session    required   pam_env.so readenv=1 envfile=/etc/default/locale

# Enforces user limits defined in /etc/security/limits.conf
session    required   pam_limits.so

# The standard Unix authentication modules
@include common-auth
@include common-account
@include common-session
```

However, in some situations, passwords are important for more than just authentication. The most common is when requesting a Kerberos ticket with `pam_sss.so` and when mounting an encrypted, remote drive with `pam_mount.so`.

For these scenarios Posit Workbench supports an optional mode to retain passwords after a PAM-based login and then forwards them on to PAM sessions. This is enabled via the `auth-pam-sessions-use-password` setting:

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

```
auth-pam-sessions-use-password=1
```

When this setting is enabled, you must also remove the use of `pam_rootok.so` from the PAM profile – otherwise modules (like `pam_mount.so`) in other stages may not see the password, even if it has been provided.

To use the Job Launcher, more settings are required:

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

```
launcher-sessions-enabled=1
launcher-use-ssl=1
auth-pam-sessions-enabled=1
```

> Note
>
> This configuration demands that Posit Workbench retain user passwords in memory. This retention is done securely using industry best-practices.

More resources

To learn more about PAM profile configuration, the following are good resources:

- http://www.linux-pam.org/Linux-PAM-html/Linux-PAM_SAG.html
- http://linux.die.net/man/8/pam.d
- http://www.linuxjournal.com/article/2120
- http://www.informit.com/articles/article.aspx?p=20968

PAM sessions with Kerberos

> Important
>
> Managing Kerberos tickets in this manner requires enabling PAM-based password forwarding. You must also create a custom profile; the default `/etc/pam.d/su` profile will not work with Kerberos.

PAM sessions can be used in conjunction with SSSD and the `pam_sss.so` module to grant Kerberos tickets for IDE sessions. Installation of these components varies depending on the platform. Additionally, other third-party PAM modules that grant Kerberos tickets can be made to work in the same general way, but this is out of the scope of this guide.

As a prerequisite, you need entries similar to the following in your `rserver.conf` settings:

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

```
auth-pam-sesions-enabled=1
auth-pam-sessions-profile=rstudio-session
auth-pam-sessions-use-password=1
```

Secondly, the PAM profile used for sessions – `rstudio-session` in this example – must be updated to include `pam_sss.so` directives:

Listing 9 `/etc/pam.d/rstudio-session`

```
auth        required      pam_sss.so
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
password    sufficient    pam_sss.so use_authtok
session     requisite     pam_sss.so
```

Note

If you are migrating your Kerberos settings from the now-deprecated `pam_krb5` to `pam_sss`, consult the pam_krb5 migration documentation for additional information.

❗ Important

The PAM profile for sessions **must not include a `pam_rootok.so` directive** – you need to ensure that authentication is done by Kerberos using an explicit password.

Finally, configure SSSD itself to work with your Kerberos environment:

Note

The sample configuration above is a very simple one, but SSSD supports a large number of additional options, some of which may be required to get Kerberos working correctly in your environment. Reference SSSD's documentation before proceeding to ensure you've specified all options correctly.

PAM session cleanup

By default, Posit Workbench does not close PAM sessions when their associated IDE or home page session exits because PAM sessions can initialize and maintain resources with complex lifetimes. For example, if a user has multiple active IDE sessions then closing the PAM session associated with one of them might unmount a drive or revoke a Kerberos ticket that is still required by another running IDE.

We caution against relying on PAM sessions closing since this is not a reliable mechanism to clean up resources.

To force Workbench to close PAM sessions you can use the `auth-pam-sessions-close` setting:

Listing 10 `/etc/sssd/sssd.conf`

```
[sssd]
services = nss, pam

# replace this with a comma-separated list of your configured SSSD domains
domains = TEST.EXAMPLE.COM

[domain/TEST.EXAMPLE.COM]
# can also be set to ad or local depending on your authentication setup
id_provider = ldap

auth_provider = krb5

# replace with the name of your Kerberos realm
krb5_realm = TEST.EXAMPLE.COM

# we recommend setting the debug level high to make troubleshooting easier
debug_level = 5

krb5_validate = true

# note that RHEL-7 default to KERNEL ccaches, which are preferred in most cases to FILE
krb5_ccachedir = /var/tmp

krb5_keytab = /etc/krb5.keytab
```

> ⚠ Warning
>
> IDE sessions do not respect this setting when the Job Launcher is enabled. Only the user's home page closes its PAM session.

If you are certain that you need PAM sessions closed, disabling support for running multiple IDE sessions avoids potential conflicts.

Testing and troubleshooting

PAM sessions can be tested outside of Workbench using the `pamtester` utility described in Diagnosing PAM authentication problems. For example:

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

```
auth-pam-sessions-close=1
```

```
sudo /usr/lib/rstudio-server/bin/pamtester --verbose \
  <session-profile> <user> authenticate acct_mgmt setcred open_session
```

Substitute the actual session profile name (`su` by default) for `<session-profile>` and an actual username for `<user>`.