Modifying the loginwindow Application

Viewing the loginwindow Mechanism

To list the currently installed loginwindow mechanism on a computer, execute the following command:

```
security authorizationdb read system.login.console
```

XML in PLIST format similar to the following should display:

```
<key>mechanisms</key>
<array>
<string>builtin:policy-banner</string>
<string>JamfConnectLogin:CheckAzure</string>
<string>JamfConnectLogin:CheckOIDC</string>
<string>JamfConnectLogin:PowerControl,privileged</string>
<string>JamfConnectLogin:CreateUser,privileged</string>
<string>JamfConnectLogin:DeMobilize,privileged</string>
<string>builtin:login-begin</string>
<string>builtin:reset-password,privileged</string>
<string>builtin:forward-login,privileged</string>
<string>builtin:auto-login,privileged</string>
<string>builtin:authenticate,privileged</string>
<string>PKINITMechanism:auth,privileged</string>
<string>builtin:login-success</string>
<string>loginwindow:success</string>
<string>loginwindow:FDESupport,privileged</string>
<string>HomeDirMechanism:login,privileged</string>
<string>HomeDirMechanism:status</string>
<string>MCXMechanism:login</string>
<string>CryptoTokenKit:login</string>
<string>loginwindow:done</string>
<string>JamfConnectLogin:EnableFDE,privileged</string>
<string>JamfConnectLogin:SierraFixes,privileged</string>
</array>
```

The Mechanisms key lists the loginwindow settings as an array of strings. Mechanisms defined as "privileged" prompt the loginwindow to run the mechanism as the root user. The only built-in macOS mechanism removed by Jamf Connect is `loginwindow:login`, which displays the standard macOS login window.

Editing loginwindow Settings

You can use the authchanger binary that is installed with Jamf Connect Login to edit loginwindow settings. For more information, see authchanger.

You can also edit loginwindow mechanisms manually on macOS:

1. Open Terminal, and execute the following command:

   ```
   security authorizationdb read system.login.console
   ```

2. Using your preferred text editor, edit the XML mechanism array.
3. Reload the list using the security command executed as root:

   ```
   sudo security authorizationdb write system.login.console < newest.xml
   ```

> **Note:** If the loginwindow application is running, you must restart it to apply your changes. If no users are currently signed in to the computer, you can close the loginwindow application as root with the following command: `sudo killall loginwindow`

If a user is currently logged in to the computer, the user must log out.

Additionally, it may be useful to leave an admin user signed in to a Finder session, and then Fast User Switch to the loginwindow. Using the `killall` command above will kill any Finder sessions currently running, including your admin users.

> **Note:** Fast User Switching must be enabled on the computer to use this feature. Navigate to **System Preferences** > **Users & Groups** > **Login Options** to access this feature.

Disabling and Re-enabling the Jamf Connect Login Window

You can disable and re-enable the Jamf Connect login window by executing the following commands:

Disable: `authchanger -reset`

Re-enable: `authchanger -reset -OIDC`

> **Notes:**
>
> - When disabled, the default macOS loginwindow will be displayed to users.
> - If using Okta, replace "`-OIDC`" with "`-okta`"

Restoring the Authorization Database

If the Jamf Connect login window is not loading and other remediation steps have not worked, you can replace the authorization database by renaming the auth.db file and letting the system replace it with a default copy.

1. Start the computer in recovery mode.
2. Rename the authorization database file by executing the following command:

   ```
   mv /var/db/auth.db /var/db.auth.db.bak
   ```

3. Log out and let the computer finish startup.

The standard macOS login window should appear.

‹ Testing Cloud Authentication

Verifying Local Account Creation ›